Click below to quickly navigate to Sections and Appendices

System Development Methodology

Release 6.01

U.S. Department of Housing and Urban Development

January 2000

For information or comments please send e-mail to the SEO&PMD mailbox

TABLE OF CONTENTS

Return to Title Page

FIGURES

TABLES

Return to Title Page

Introduction

This page intentionally left blank.

INTRODUCTION

Purpose

The Department of Housing and Urban Development (HUD) System Development Methodology (SDM) provides an approach for solving information management needs that arise during the lifecycles of automated information systems. Although this methodology should be used on all information system projects related to HUD programs, it is intended to be flexible so that it can be tailored to meet the needs of each project.

The objective of every HUD information system project is to construct software products that are engineered to satisfy the user's requirements, within determined cost, schedule, and quality guidelines. These information system projects encompass new development, maintenance, reuse, reengineering, and all other activities resulting in software products. The software products have a wide spectrum of characteristics:

- Target operating environments include personal computers (PCs), mainframes, and LAN-based and client/server-based systems.
- Product sizes range from only a few thousand lines of code to more than a million.
- Cost and cycle-time requirements vary.
- Desired end-product qualities, such as reusability and reliability (consequences of software failure ranging from minor inconvenience to loss of several million dollars), also vary.

One significant lesson learned from many years of information system development throughout HUD is that no single solution can solve every problem. No one lifecycle model; analysis and design, testing, or product evaluation method; or degree of formality for documents and reviews is appropriate for all HUD system projects. To accommodate these variations, the project manager of each project must shape its system process to acknowledge customer requirements and constraints; goals and objectives for cost, cycle time, and product qualities; and management's tolerance for risk.

This SDM has the following objectives:

- Clarify the importance, objectives, and benefits of system lifecycle management to all potential participants in the system lifecycle.
- Describe the progression of the lifecycle through individual activities and processes, in terms of their respective objectives and products, and describe the relationships among the activities.
- Guide organizations to comply with level 2 of the Software Engineering Institute (SEI) Capability Maturity Model (CMM).

This SDM presents a methodology applicable to the development and maintenance of all HUD information system projects. Systems that support HUD programs vary greatly in size, scope of application, complexity of processing, technologies used, and the methods and tools employed to support the evolution of the system from initial need statement through operation and eventual system termination. To accommodate the diversity of system development needs in HUD programs, this methodology offers a structured, disciplined approach.

Benefits Gained from Using HUD's SDM

An SDM represents many years of experience and lessons learned by information management professionals. The HUD SDM builds on this experience, making use of other organizations' experience and guidance as well as HUD's own experience. It is intended to ensure that each information system project is successful and avoids the pitfalls of learning, or relearning, the lessons of information systems through trial and error. This SDM is expected to accomplish the following:

- Ensure ongoing evaluation of the HUD program environment and associated system and data requirements, from project approval through the entire lifecycle of the system.
- Provide early identification of technical and management issues and avoid investment in impractical or unfeasible system features.
- Identify, early in the lifecycle, the total resources needed, including those required for development and operational support throughout the life of the system.
- Consider full costs of the system during the decision making process.
- Foster realistic expectations of system capabilities in the user community.
- Clarify the extent to which proposed system modifications will affect programmatic, technical, management, and cost elements of the overall system.
- Encourage periodic evaluations to identify systems that may no longer be needed or may be too expensive to continue using.
- Provide clear measures of system progress and status to anticipate a possible need for corrective action.
- Supply much of the data needed to support information resources management planning and budget requests.

HUD SDM Lifecycle Models

The SDM acknowledges that a single lifecycle model is not appropriate for all HUD system development projects and, if necessary, a project may use multiple lifecycle models to accomplish the project's objective. (For example, a waterfall development lifecycle model may be the best overall model to use on a specific project, but an evolutionary lifecycle model may be inserted in the overall lifecycle to prototype a high-risk area.) The SDM also acknowledges that the appropriate method for accomplishing a required activity depends on characteristics of the system project.

The activities defined in the SDM can support various lifecycle models, such as waterfall development, incremental development, evolutionary development, package-based development, and legacy system maintenance.

Software Development Approach

The HUD software development addresses a broad range of activities, from identifying a need for building or acquiring a system to the final disposition of the system.

The HUD SDM is intended to provide the flexibility to select the activities, methods, tools, and technologies that are appropriate to solving each information management need and to permit selection and refinement of the basic lifecycle as appropriate for a given need. However, this flexibility exists within a framework that requires the active participation of HUD program managers and information management professionals throughout the lifecycle.

The HUD system development methodology consists of the following six phases:

- Initiate Project
- Define System
- Design System
- Build System
- Evaluate System
- Operate System

Following is a summary of each phase:

- Initiate Project involves (1) identifying the information management need and (2) deciding whether to commit the resources necessary to address the need.
- Define System involves developing a detailed functional statement of the user's need and developing a project plan covering the estimated cost, schedule, and technical parameters of the project.
- Design System involves developing and approving detailed specifications to fulfill the stated functional requirements.
- Build System involves developing and testing computer programs in response to approved specifications and preparing documentation needed for the ongoing operation and maintenance of the system.
- Evaluate System involves testing of the system to ensure that the system functions as desired and certifying that the system has completed development.
- Operate System involves installing the system at all sites and the onset of operational activities, including controls on all proposed computer hardware and software changes.

The duration of each phase and associated levels of effort will vary considerably from system to system. Using systems analysis, design, or development tools tends to reduce the duration of the activities in which the tools are used.

Tailoring the Software Development Approach

For some information management needs, it may be desirable to alter the software development approach. For very simple systems, such as single-user applications, it may be appropriate to combine parts of activities. For projects using iterative methods, such as rapid prototyping, some activities may overlap. For very large or complex systems, it may be appropriate to divide the system into major subsystems or other partitions and manage the evolution of each subsystem through its own development approach. For these very large or complex systems, it will be important to develop project team structures and other mechanisms to ensure effective coordination across subsystems. In adapting the development approach to suit a particular situation, the following considerations are of particular importance:

- Document clearly in the Project Plan any tailoring of the system development approach.
- Identify early and explicitly (i.e., not by default or accident) issues regarding project approach, execution, and continuation; resolve these issues no later than the completion of the corresponding activity.
- Include system reviews and approvals in any tailoring of the system development approach to ensure appropriate program management participation and oversight; as appropriate, combine reviews and approvals to reflect consolidation of individual activities.

When determining the path that the development effort will take, managers must consider security, efficiency, cost effectiveness, and compatibility with HUD's operating environment. In particular, managers should weigh potential savings in operation and maintenance of the system against one-time, short-term savings that may occur from tailoring certain development activities.

Crosscutting Considerations for All SDM Phases

A number of considerations exist throughout system development for every system and are addressed in multiple activities. These considerations are summarized as follows:

- **Project Plan.** The Project Plan is produced during the Initiate Project phase and is updated, expanded, and refined continually throughout system development. It covers project scheduling, work breakdown structure (WBS), staffing, resources, adjustments to the system development cycle structure, selection of tools and techniques, identification of applicable reviews and approvals, configuration management (CM) methods, and other related topics. The plan is subject to review and approval by HUD management.
- **Project Reviews.** Project reviews are conducted at each phase of system development to ensure that the system ultimately established is programmatically and technically sound. Reviews help to ensure that key issues are identified and addressed appropriately as early as possible in the system's development to avoid major, expensive rework in later activities. Reviews provide feedback to the project team and also assist management in supporting required system approvals. Specific organizations and individuals who are to participate in project reviews and related activities are designated early in the lifecycle and are selected in light of the nature of the information management need and the recommended solution.
- **Quality Assurance (QA).** During the early stages of a project, the QA group works with the project development team to establish plans, standards, and procedures that will add value to the software project and satisfy the constraints of the project and the HUD Information Technology (IT) organization's policies. By participating in the early stages in establishing these guidelines, the QA group helps ensure that the plans, standards, and procedures fit the project's needs and verifies that they will be useful for performing reviews and audits throughout the software lifecycle. The QA group continuously reviews project activities, audits software work products throughout the lifecycle, and provides management information with which to judge whether the software project is adhering to its established guidelines. The QA group reviews and audits software products (software and documentation) and process activities to verify that these products comply with the

SDM and the applicable procedures and standards. The QA group provides the software project and other appropriate managers with the results of these reviews and audits. A sound QA program ensures that the project is following approved HUD standards and procedures and that quality is checked throughout the product lifecycle.

- **Project Approvals.** Formal approvals also occur at each phase in the lifecycle to confirm management support of the project and the resulting system. Conducting reviews and obtaining approvals is not the goal of the lifecycle process, but they are key means to the desired end: successfully solving an information management need. Program management approvals are obtained during all activities from project approval through operation. In selecting organizations and individuals to provide approvals for a particular project, consideration is given to the specific characteristics of the information management need and the proposed solution.

- **Configuration Management (CM).** Continual, consistent documentation of the development and evolution of the system is essential to ensure that all milestones are met, key decisions are recorded, the system can be accurately described, and a consensus is reached on what products are required for delivery. CM must maintain a controlled library of project products, software, hardware, and documentation and provide a process for the consideration and disposition of requested modifications to the system.

- **Data Administration.** To ensure effective management of HUD's data resources, all systems are created and maintained in accordance with HUD's data administration policy and practices. Project activities are carried out consistent with the data management environment, and data administration concerns are addressed during all activities of the lifecycle.

- **Methods, Techniques, and Tools.** All systems created and maintained by HUD use clearly articulated methods, techniques, and modern system development and maintenance tools to the greatest extent practical. Specific selections are confirmed in subsequent activities as appropriate. However, no methods or tools can replace the application of system development management. For example, prototyping may be used as a method of performing design or development tasks, but the final products always include the documentation and reviews prescribed by the development approach. Use of expert system development tools may require that some activities be performed in a different sequence or iteratively; but, again, they are documented and reviewed as prescribed by the lifecycle management model.

Key HUD SDM Principles

The following principles serve as the foundation for managing the system development approach within HUD:

- **Ensure a structured approach to solving information management needs.** HUD's SDM describes an overall approach for achieving solutions to identified information management needs. Primary emphasis is placed on the decisions that need to be made and the proper timing of these decisions. This SDM provides the flexibility for tailoring the approach, enabling information system projects to combine activities, processes, and products as appropriate and to select the tools and methodologies best suited to solving the need.

- **Designate an accountable sponsoring organization for each system project.** The project sponsor helps to ensure effective planning, management, and commitment to information system projects. The sponsor serves in a leadership role, providing guidance to the project team and securing from senior management the required reviews and approvals at specific points in the lifecycle.
- **Appoint a single project manager for each system project.** The project manager has lead responsibility for the project's success and works through a project team and other supporting organization structures (e.g., working groups and user groups) to accomplish the objectives of the project. Regardless of organizational affiliation, the project manager is responsible for ensuring that project activities and decisions consider the perspectives of all affected organizations.
- **Require a comprehensive project plan for each system project.** The project plan is a pivotal element in successfully solving an information management need. The document describes how the system development approach will suit the project's specific characteristics. It also provides direction to the many activities of the lifecycle. It is developed in its initial form during the Initiate Project phase and is refined and expanded throughout the system lifecycle.
- **Assign specific individuals to perform key roles throughout system development.** Certain roles are considered vital to a successful system project, and at least one individual must be assigned to fulfill each role on a full- or part-time basis, as appropriate. These roles include program management, program staffing, QA, and CM. For most projects, more than one individual should represent the actual or potential users of the system (e.g., program staff) and should be designated by the program managers of the affected programs and organizations.
- **Obtain the participation of skilled individuals.** Individuals responsible for conducting system lifecycle management projects are encouraged to obtain assistance from experienced information management professionals. The skill of the individuals participating in a system project is the single most significant factor for a successful project. This SDM is not intended as a substitute for information management skills or experience. Rather, it was developed to provide an environment in which skilled personnel could perform to the best of their abilities.
- **Document completely and accurately activity results and decisions.** Effective communication and coordination of activities throughout system development depends on complete and accurate documentation of decisions and activities leading up to decisions. Activities and decisions should not be considered complete until there is tangible documentation of the activity or decision.
- **Emphasize data management throughout system development.** Data processed by a system is a valuable agency resource; therefore, this methodology stresses effective data management, conducted in accordance with Federal guidelines. Because of the large volume of data handled by HUD systems, the increasing trend toward sharing data across systems and programs, and the importance of data quality, it is essential that the selected system lifecycle activities stress the clear definition of data and the design and implementation of automated and manual processes to ensure proper safeguarding of the system's data.
- **Review and formally approve each system project.** To help ensure that systems effectively address the targeted information management need, each project is

subject to management review and approval. Skilled professionals should perform the reviews by examining tangible products from a programmatic, technical, and project management perspective. Reviews aid the project team and those who provide the required project approvals. Approvals are provided by the suitable level of management and confirm the continued commitment to the project scope, direction, and resource requirements in view of known risks and uncertainties.

- **Clarify resource availability before a system project proceeds.** Beginning with the approval of a project, the continuation of a system project is contingent on a clear commitment from the sponsoring management. This commitment is embodied in the assurance that the necessary resources will be available, not for the next activity only, but for the remainder of the lifecycle.
- **Assure, within reason, that information processed by the system is reliable and properly safeguarded.** Proper precautions should be followed throughout development, testing, and operation of the system to ensure that data processed by the system is accurate and that requirements of the Federal governing directives are being met.

Software Engineering Institute Capability Maturity Model

The Software Engineering Institute (SEI) Capability Maturity Model[1,2] (CMM) for software is a framework that describes the key elements of an effective software development process. This model discusses practices for planning, engineering, developing, and managing software development and maintenance projects. The CMM describes an evolutionary improvement path from an ad hoc, immature process (Level 1) to a mature, disciplined process (Level 5).

The SDM has been updated to supplement IT system development practices and assist the organization in reaching CMM Level 2, the Repeatable level. An objective of Level 2 is to institutionalize effective software management processes for software projects, allowing organizations to repeat successful practices developed on earlier projects.

HUD SDM Document Organization

The HUD SDM document consists of this introduction, six major sections that describe the major phases, and additional supplemental appendices.

The Introduction presents the purpose and benefits of the SDM. An important concept of this SDM is that it can be tailored to satisfy the requirements of any HUD system development project, providing every project a documented system development methodology.

Section 1, Initiate Project, describes the activities required to identify information management needs and the activities to decide whether to commit resources to satisfy or solve a need.

Section 2, Define System, describes processes used to establish and document the system's functional and data requirements.

---

[1] M.C. Paulk, B. Curtis, M.B. Chrissis, C.V. Weber, *Capability Maturity Model for Software, Version 1.1,* Software Engineering Institute, CMU/SEI-93-TR-24, February 1993.

[2] M.C. Paulk, C.V. Weber, S. Garcia, M.B. Chrissis, and M. Bush, *Kay Practices of the Capability Maturity Model, Version 1.1,* Software Engineering Institute, CMU/SEI-93-TR-25, February 1993.

Return to Title Page

Section 3, Design System, discusses developing a system design from the system functional and data requirements.

Section 4, Build System, presents the process used by the development team to build and test the system before releasing it to the test team.

Section 5, Evaluate System, discusses the process used by the test team to test the system to ensure the user community that the system functions as expected.

Section 6, Operate System, gives operational guidelines for system user activities, including system installation and operation and reporting of system deficiencies and enhancements.

Appendix A, Project Management Guidelines, presents a set of guidelines for project managers to use in planning and managing a project. Key project planning concepts include defining the scope of the project; selecting the best lifecycle model; organizing the work in manageable units; estimating the time to complete each unit of work; assigning staff to perform each work unit; and organizing the project development team to include independent test, CM, QA, and other support personnel.

Appendix B, Configuration Management Guidelines, presents the basic philosophy and activities required to implement a successful configuration management program.

Appendix C, Quality Assurance Guidelines, presents the philosophy and activities required to implement a sound quality assurance program.

Appendix D, Maintain System, outlines activities used to implement any changes to a system after it is installed and released to the production environment.

Appendix E, Document Review Checklists, presents minimum criteria for project products.

Appendix F, Work Breakdown Structure, presents an introduction to the fundamentals of HUD IT work breakdown structure (WBS) and its application to the work efforts defined in the HUD IT environment.

Appendix G, Software Prototype Methodology, presents the activities for prototyping, the interfaces for iterative development, and the role of a prototype effort within the context of the SDM.

Table 1 is a description of SDM chapters and recommended reading for specific audiences.

Return to Title Page

Table 1. Which SDM Chapters Should You Read?

Primary Audience	Secondary Audience	Chapter	Title	Summary
All		Introduction	Introduction	Provides a summary of the philosophy behind a system development methodology and the reason why system development teams should use a disciplined approach to develop systems.
Development Team	Management	1	Initiate Project	Describes the activities used to identify the information management need and to decide whether to commit re-sources to resolve the need.
Development Team and Data Administration	Management	2	Define System	Describes the activities used to analyze and document the system and data requirements.
Development Team and Data Administration	Management	3	Design System	Describes the activities used to develop the design of the database and computer programs.
Development Team	Management	4	Build System	Describes the activities used to develop the database and computer programs.
Test Team	Management	5	Evaluate System	Describes the activities used to independently test the system before release to the production environment.
Development Team/ Computer Services	Management	6	Operate System	Describes the activities used to install and implement a system.
Management	Development Team	Appendix A	Project Management Guidelines	Presents a set of guidelines for a project manager to use in planning and managing a project.
Configuration Management	Management/Technical	Appendix B	Configuration Management Guidelines	Presents a set of guidelines to configure and control a system environment.
Quality Assurance	Management/Technical	Appendix C	Quality Assurance Guidelines	Presents a set of guidelines to establish a QA program.
Development Team	Management	Appendix D	System Maintenance	Describes the activities used to perform system maintenance.
Development Team	Management	Appendix E	Document Review Checklists	Presents minimum criteria for project products.
Management		Appendix F	Work Breakdown Structure	Describes how to apply WBS in the HUD IT environment.
Development Team	Management	Appendix G	Software Prototype Methodology	Describes activities for prototyping

Return to Title Page

Section 1. Initiate Project

1.0 Initiate Project Phase

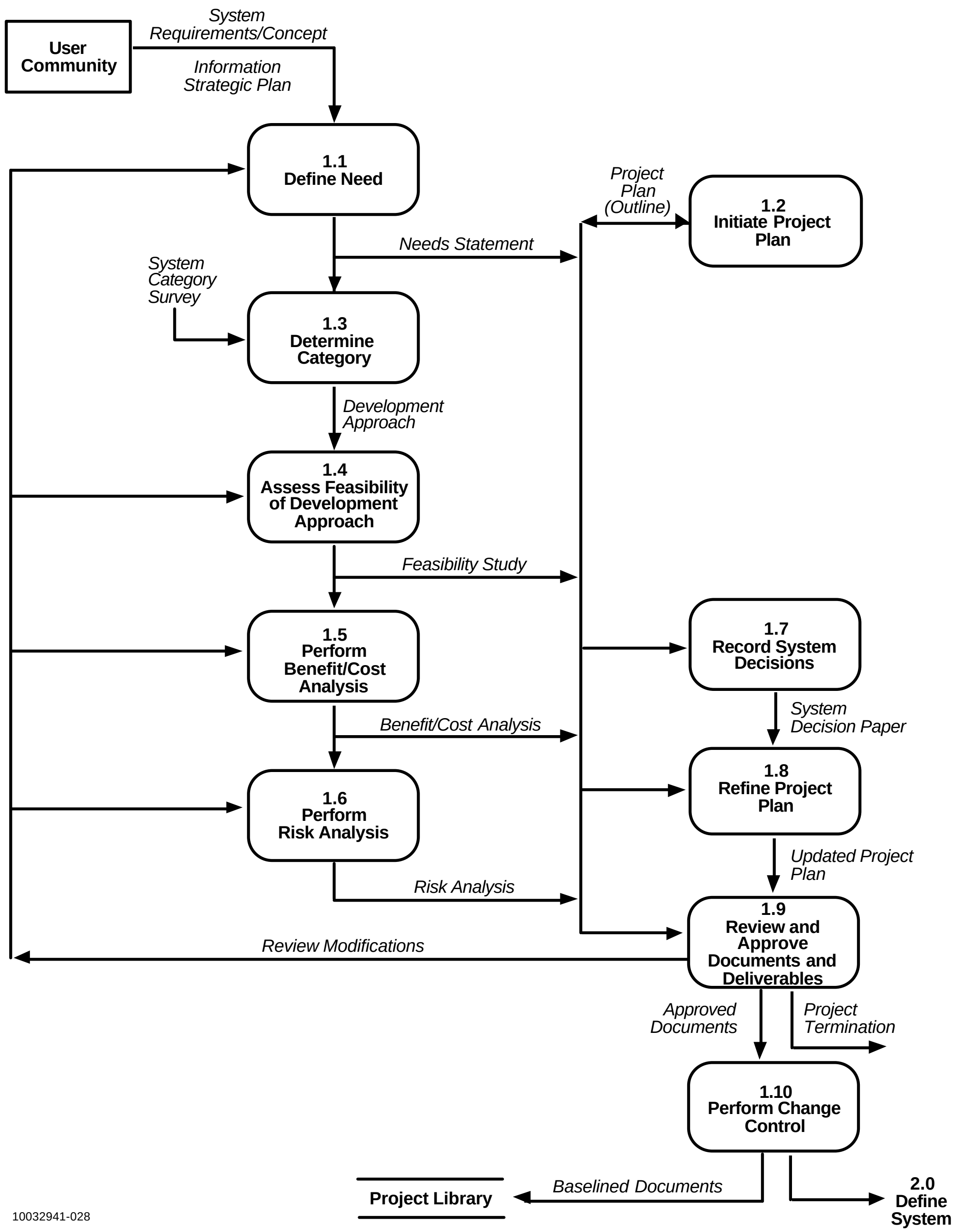

Figure 1-1. Process Flow for Initiate Project Phase

1.0 INITIATE PROJECT

Purpose

The Initiate Project phase of system development is the period in which an information management need is identified and the decision is made whether to commit the necessary resources to solve the deficiency. Figure 1-1 highlights the process flow for the Initiate Project phase.

Overview

The activities of the Initiate Project phase begin with identification of an information management need. A user or system sponsor may identify the need or it may be the result of the strategic information architecture analysis described in the HUD Information Strategy Plan (ISP). The need is documented in the form of a Needs Statement or an Advanced Requirements Notice (ARN). These documents describe the deficiency and justify the exploration of alternative solutions. Large-scale development and major system enhancements are described in a Needs Statement by the project sponsoring organization and approved by the Technology Investment Board (TIB). Smaller development efforts, minor enhancements, or maintenance efforts are described in an ARN and approved at the group or division level. An Evaluation of Request (EOR) is used to document the evaluation of the ARN.

After a need has been fully documented and approved, the project manager plans the Initiate Project phase activities. The project manager evaluates the scope of the project and documents high-level phase activities in a preliminary Project Plan. Decisions regarding size, complexity, and effort are to be made on a project-by-project basis. The project manager classifies the proposed development effort as a new development or as an enhancement to an existing system. The applicable selection criteria have been designed to provide assistance when determining that classification. The project manager further categorizes the project as either a major or a minor effort according to its size and complexity. In addition, the project manager makes decisions concerning the analysis, design, and development techniques and tools for the project, including computer-aided software engineering (CASE) tools.

After an information management need has been described, documented, and approved, alternatives for solving the need must be identified. Criteria for evaluating these alternatives are developed based on the objectives of the proposed project. Each alternative is evaluated for its technical, cost, and schedule feasibility, and a determination is made as to the best development option. The steps of this analysis are documented in a feasibility study that describes the criteria used to evaluate each alternative and identifies the preferred development approach.

For each alternative considered, a benefit/cost analysis is performed. All economic benefits and costs, including development and operation, one-time, and recurring costs, are calculated in current dollar values. Benefits that cannot be quantified in dollar amounts are identified. Each alternative is then evaluated based on the net economic benefits of developing and operating the system, as calculated during the analysis. This evaluation is documented in the benefit/cost analysis, including the analysis methodology, the alternatives with their associated benefits and costs, and a recommendation as to the best development alternative.

A risk analysis on system security is then performed, identifying the strengths and weaknesses of the proposed project as well as alternatives for improvement. The purpose of the risk analysis is to select cost-effective safeguards that will reduce security-related risks to an acceptable level. The following determinations are included in a risk analysis document:

- Type of risk analysis to be performed, based on the category determination made earlier in the phase
- Security requirements, based on the sensitivity and criticality of the proposed project to the project sponsor and user organizations
- Baseline security requirements that must be satisfied to ensure that applicable HUD guidelines are met
- Threats to the proposed project and the vulnerabilities that could be exploited
- Countermeasures to the identified threats and vulnerabilities

All decisions made during the Initiate Project phase are recorded in a System Decision Paper (SDP), as is the schedule of activities already accomplished. In addition, a projected schedule of activities for the full lifecycle of the system, by phase, is provided in the SDP. Included in this schedule are the actual or projected beginning and ending dates for each phase, as well as a running total of the actual or projected development and operation costs, broken out by phase and totaled by fiscal year. Major, complex development efforts, generally those addressing departmental priority needs, should follow the activities and tasks as outlined in the System Development Methodology (SDM). Minor efforts may need to tailor the SDM to fit the project's specific needs.

Based on the findings summarized in the SDP, the appropriate review board decides whether to continue with the proposed development effort. If the review board decides to continue, the project manager documents the decision and which development alternative is selected in the SDP. The project manager then updates the Project Plan accordingly.

Reviews of the project are performed by one or more of the following groups:

- **Technology Investment Board (TIB)**. Responsible for ensuring that Departmental resources are directed to the highest Information Resource Management (IRM) priorities of the Department and that these limited resources are efficiently utilized. The TIB tracks and reviews progress toward achieving key events and target dates for critical project activities.
- **Change Control Board (CCB)**. Serves as the decision-making body for each program area project. The CCB is the control mechanism for the program office that has requested the need for which the project has been initiated. For each project, the project sponsor establishes and heads the CCB. The project sponsor appoints CCB members from the following organizations: program officials, system users, external stakeholders, and IT representatives.

  The project CCB evaluates the scope, applicability, and effect of each requested requirement change (RC). The CCB focuses on items that could affect cost, schedules, or compliance with technical requirements. It acts on any requested RC to the system and provides change approval or disapproval based on defined strategic initiatives, program business objectives, and budgetary parameters. In addition, the project CCB meets regarding impacts of changes proposed by other program areas or organizations, especially schedule and cost impacts. The project CCB has the authority to establish project baselines, initiate or change software, accept testing results, and approve the release of software into production.
- **Quality Assurance (QA)**. Responsible for coordinating and implementing software quality assurance for IT projects. QA verifies that plans, standards, and procedures

Return to Title Page

are in place and can be used to review and audit the project. In addition, QA reviews software development and maintenance activities to verify compliance. Results of QA activities are reviewed with IT management and project personnel.

After approval, all documentation is placed under the configuration management change control process. This change control process is observed whenever any products of the project's baseline are revised. Change control activities include verifying changes made to the products, assigning a new version number to the revision, updating a log book with the change information, updating the central library with the new version, distributing copies of the new version, and archiving the old version. This process of change control over the lifecycle products continues throughout each project phase.

After all deliverables have been completed (see Table 1-1), the level of review and approval required for the project is determined based on management guidelines. The Initiate Project phase culminates in a decision as to whether to continue the project and proceed to the next phase of the lifecycle, the Define System phase.

Table 1-1. Initiate Project Phase Functions and Products

Initiate Project Phase Function	Product
1.1 Define Need	Needs Statement
1.2 Develop Project Plan	Project Plan
1.3 Determine Category	--
1.4 Assess Feasibility of Development Approach	Feasibility Study
1.5 Perform Benefit/Cost Analysis	Benefit/Cost Analysis
1.6 Perform Risk Analysis for System Security	Risk Analysis
1.7 Record System Decisions	System Decision Paper
1.8 Refine Project Plan	Project Plan (updated)
1.9 Review and Approve Documents and Deliverables	New products Management summary[1]
1.10 Establish and Perform Change Control Activities	Change control records

[1] A management summary is prepared for each product produced or revised during the Initiate Project phase. This one-page summary includes a summary of the essential data collected in a document product, conclusions that may be drawn from the document, and potential impacts on the project, if applicable.

Standards and Guidelines

HUD SDM documentation standards (Handbook 2400.15) and Project Management Guidelines (Appendix A) should be followed during development of Initiate Project phase products. For the benefit/cost analysis, use the HUD, Benefit/Cost Analysis Methodology: Volume I - Methodology, and Volume II - Workbook, September 1995.

Roles and Responsibilities

Throughout the Initiate Project phase of development, key personnel are required to perform the various tasks and activities outlined in the SDM. Table 1-2 presents the types of personnel required and the activities for which they are responsible.

Return to Title Page

Table 1-2. Roles and Responsibilities for Initiate Project Phase (1 of 2)

Role	Responsibility
Project Sponsor	Establishes and heads project CCB for program office. Coordinates review and approves Needs Statement or ARN. Reviews and approves deliverables. Reviews and approves system development decisions. Ensures compliance with the Department Financial Management Systems Strategic Integration Plan (DFMSSIP).
Project Manager	Records need and system category. Makes decision on system development approach. Develops Project Plan. Selects project development team. Manages development of deliverables. Coordinates system development decisions with project sponsor. Updates Project Plan to include actual costs incurred and specific activities accomplished for the Initiate Project phase; and revises project schedule, plans, strategies, resources, and requirements for the Define System phase, as required. Reviews, for approval, the quality process planned by users and developers for the project phases. Reviews and approves deliverables prior to submittal. Obtains appropriate approval for deliverables. Provides quarterly status reports to TIB.

Return to Title Page

Table 1-2. Roles and Responsibilities for Initiate Project Phase (2 of 2)

Role	Responsibility
Project Development Team (User and Developer)	Identifies need and prepares Needs Statement or ARN. Determines system category. Determines development options. Prepares system category survey. Performs risk analysis. Conducts feasibility study. Performs benefit/cost analysis. Provides input to project manager on required Define Project phase tasks and activities. Participates in review of Project Plan. Develops SDP. Identifies and plans the quality process for the project phases.
Computer Services	Provides input to risk analysis, feasibility study, and benefit/cost analysis. Initiates capacity planning estimates. Provides input for system category options.
ADP Security	Reviews phase products to ensure that security requirements have been incorporated. Reviews and approves risk analysis.
Project Data Administrator	Reviews new projects for adherence to HUD information architecture.
Configuration Management	Establishes Configuration Control Board (CCB). Develops Configuration Status Accounting (CSA) function. Establishes a central Project Library. Establishes change control for new products. Performs change control for new products. Provides status reports to project management as requested or required.
Quality Assurance	Assists in all facets of review activities. Reviews deliverables for adherence to applicable HUD documentation standards and project guidelines. Coaches users and developers in the planning for quality in the project phases.

1.1 DEFINE NEED

The information management need represents any of multiple requirements: a functional capability that is not met by existing systems, increased or streamlined workflow, manual-to-automated system conversion, or productivity improvement. The need may be identified by clients or system sponsors, or it may be a result of the analysis documented in the HUD Department ISP. The deficiency to be satisfied is identified as narrowly as possible to increase the probability of developing or modifying a system that will accurately meet the user's requirements. During the process of identifying the need, the organization(s) affected by the need is identified. The conditions that restrict resolution of the need and the estimated cost of fulfilling the need are determined. Activities include the following:

1.1.1 Identify the need.

1.1.2 Document the need.

1.1.3 Obtain approval of a Needs Statement or ARN.

All data regarding the information management need are documented in the form of a Needs Statement or an ARN, which is prepared by the user and approved by the project sponsor. The project sponsor also certifies the intention to provide project funding.

The Needs Statement or ARN is subject to the project's quality assurance process to ensure that it includes the necessary level of detail and completeness and has been developed to meet HUD standards. The final step in this process is to obtain approval from the appropriate HUD authority.

Table 1-3 defines roles and responsibilities of key personnel for defining the project need.

Table 1-3. Roles and Responsibilities for Defining Need

Role	Responsibility
Project Sponsor	Coordinates review of the Needs Statement or ARN. Approves the Needs Statement or ARN.
User	Identifies need and prepares Needs Statement or ARN.

1.1.1 Identify Need

The initial activity undertaken to develop a Needs Statement or ARN is the identification and description of the need. The description should be as specific as possible to ensure the development of a system that will satisfy the requirements. The following activities are performed when identifying the need to be solved.

- Identify the functional area.
- Identify, by title and function, the organization(s) with the need. If more than one organization is cited, indicate which is the primary user.
- Cite specific directives.
- Describe the current organizational and functional environment.
- Determine the scope of the deficiency.
- Describe the functions that are needed but not currently provided. Identify benefits to be expected if the needed functions are developed and implemented.
- Summarize needs in terms of outcome to be achieved.
- Determine benefits.
- Identify constraints.
- Identify conditions that restrict or hinder resolving the need, such as relative priority within the functional area; limits on available funding; interfaces with other systems; and staffing, security, or logistical considerations.
- Estimate costs.
- Estimate the costs in terms of resources required to fulfill the need. This is accomplished by estimating the costs to develop and implement the project and the costs to complete the study of the need and consider alternative solutions.
- Estimate the range of costs to develop and implement the system.

- Identify the events, tasks, and estimated resources needed to complete concept development.
- Indicate consistency with the Department's Financial Management Systems Strategic Integration Plan (DFMSSIP**).**
- If the solution to the information management need requires development of a financial management system or impacts current or proposed financial management systems, indicate consistency or inconsistency with the DFMSSIP objectives as follows:
  - If solutions are *consistent* with the DFMSSIP:
    - Reference the DFMSSIP.
    - Explain how the development effort fits into the plan's schedule.
    - Demonstrate consistency with the plan's objectives.
  - If solutions are *inconsistent* with the DFMSSIP:
    - Reference the DFMSSIP.
    - Explain the inconsistency from a timing or solution standpoint, e.g., correction of a material weakness that cannot wait until implementation of the integrated system.
    - Apply for a temporary waiver.

1.1.2 Document Need

After analysis of the need has been completed, the results are documented in the Needs Statement or ARN. Document the Needs Statement or ARN in accordance with HUD SDM documentation standards (Handbook 2400.15).

The Needs Statement or ARN describes the need and the benefits to be expected and identifies the estimated costs and resources to be expended to fully develop and implement the project. Perform the following activities when documenting the need:

- *Describe the scope of the need*. Document the user organization and its functional requirements.
- *Describe the benefits to be expected*. Document the benefits to be expected if the project is approved for development.
- *State the estimated costs and resources*. Document the costs, in dollars and number of resources, to develop the project and to complete the study of the need.
- *Reference the DFMSSIP*. Document the proposed development effort's consistency or inconsistency with the DFMSSIP schedule and objectives.

1.1.3 Review and Approve a Needs Statement or ARN

The review and approval of the Needs Statement or ARN is a process that allows all project organizations to reach concurrence on the definition of the information management need. In addition, the approval process offers senior management insight into the need. Activities include the following:

- Perform review of Needs Statement or ARN.
- Obtain approval of Needs Statement or ARN.

Perform Review of Needs Statement or ARN

Conduct a project review with project personnel and system stakeholders to ensure that the Needs Statement or ARN includes the level of detail and completeness necessary to evaluate a proposed solution to the user's information management need and that it meets the appropriate HUD and project standards and guidelines. Use the appropriate document review checklist in Appendix E to aid the review process.

Allow a minimum of 10 working days before the scheduled review, notify the personnel required to attend this review, and provide each with a copy of the Needs Statement or ARN for their pre-review. A thorough review of the Needs Statement or ARN should be held with a facilitator, noting all comments or objections that are raised during the review procedure. A consensus should be reached on one of the following decisions before the termination of the review session:

- The Needs Statement or ARN is correct and complete, as is, without further changes.
- Additional changes that need to be made are minor and do not require further review. In this case, the updates should be made and change pages distributed by an agreed-upon date.
- Required changes will have a major impact on the document. A second review must be scheduled. The changes must be incorporated and the resulting change pages distributed at least 10 working days before the second review.

Prepare a one-page management summary describing the information management need and summarizing any significant points that are addressed. Also include a brief description of any conclusions that may be drawn and their potential impact on the DFMSSIP.

Obtain Approval of Needs Statement or ARN

Present the Needs Statement for approval to the Technology Investment Board (TIB), or present the ARN to the appropriate group or division review board, at least 10 days before the scheduled decision date. Include management summary information and an approval (sign-off) record. Project approval records are maintained by the project's configuration management (CM) function. A copy is inserted into the central project library.

1.2 DEVELOP A PROJECT PLAN

After the Needs Statement is defined, the project manager establishes a high-level project plan outline for all development phases so that adequate resources are defined. Because project planning is an iterative activity, detailed planning of the subsequent phases can occur after the Initiate Project phase is completed and the project is approved for continuation. Included in the high-level plan is the preliminary strategy for handling programmatic risks that would have a direct impact on schedule, cost, and technical performance. (Appendix A provides guidelines for project planning.)

1.3 DETERMINE CATEGORY

Apply selection criteria to determine the category of the project. Overall, the selection criteria assist in determining the level of applicability for SDM activities and tasks and will provide a basis for system platform selection. The determination will be made based on several categories of factors, including size of the effort, overall scope and complexity of the proposed system, and estimated resources required to develop and maintain the proposed system. Activities include the following:

1.3.1 Apply the selection criteria.

1.3.2 Make the selection.

1.3.3 Document the selection.

The investigation leading to the category determination will create a foundation for decisions to be made in the following areas:

- Alternative analysis, design, and system development techniques
- Selection of tools to be used to develop the system
- Target platform for the proposed system
- Determination of development effort as an enhancement to an existing system or an entirely new development effort

The results of the category determination activity are documented in the System Category Survey form. The information contained in the survey records the initial system category decisions and provides a basis for describing the proposed system alternatives in the feasibility study document. Table 1-4 defines the roles and responsibilities of key personnel for determining the project's category.

Table 1-4. Roles and Responsibilities for Determining Category

Role	Responsibility
Project Manager	Reviews needs and system category and coordinates with project sponsor.
User	Analyzes needs and determines system category. Determines development options and prepares system category survey.
Developer	Assists in determining system category.
Computer Services	Provides input to determine system category.

1.3.1 Apply Selection Criteria

Specific selection criteria are applied to project requirements to classify the type of system development effort. The selection criteria are evaluated, scored, and weighed to assist in determining the following project-related factors:

- Target platform of the proposed system
- Use of analysis, design, or system development or interactive development techniques (e.g., traditional prototyping, rapid application development [RAD], joint application development [JAD])

- System classification (i.e., new development, system enhancement, system maintenance)
- Use of commercial off-the-shelf (COTS) software, with or without modifications
- Tool usage, including CASE or automated tools
- Size and complexity of effort
- Whether the system is a major or minor effort
- A decision on the level of applicability and general guidance on the portions of the SDM that must be followed

The System Category Survey (see Table 1-5) assists the evaluator in classifying the type of development effort. The survey is organized into three major areas, each with its own questions and scoring section. Each has a minimum possible score of zero (0) and a maximum score as follows:

- System Platform: 80
- Development Techniques: 64
- Type of Effort: 24

The scores for each section are designed to form a continuum for possible courses of action, rather than a yes or no answer. Very low or high scores in any section would make the strongest arguments for a particular decision. Scores midway between zero and the maximum score for a given section would indicate that either course of action has equal merit according to the criteria in the survey. The courses of action, based on the minimum zero or maximum score for each section, are as follows:

- System Platform
  - 0: Strongest case for a PC or LAN solution
  - Maximum score: Strongest case for a mainframe-based solution
- Development Techniques
  - 0: Strongest case for not using alternate development techniques such as JAD, RAD, prototyping, and CASE tools
  - Maximum score: Strongest case for alternative development techniques, including JAD, RAD, prototyping, and CASE tool use
- Type of Effort
  - 0: Strongest indication that the development effort is a maintenance effort
  - Maximum score: Strongest indication that the development effort is a new development effort

Responding to the Survey

Each question or statement and its responses are in the first cell of a row. Each statement has up to five responses. Each response is sequentially numbered. Select the statement or response that most closely answers the question as it relates to the particular development effort being studied. If a question indicates to select all responses that apply, do so. If none of the responses apply, skip the question.

At the beginning of each of the responses will be a number from 1 to 5, and in the last five columns of the survey will be corresponding numbers. Respond to the question or statement by circling or marking the number(s) in the last five columns that correspond to the most correct response(s).

Note: Survey factors may not exist for one or more columns for a given question. This distribution of factors to columns is designed to facilitate the weighted scoring process.

Scoring the Survey

After all questions have been answered for a particular area, count the number of responses for each column and enter the total number of responses in the Response Count row. Use the number in the Multipliers row, located directly below, and multiply the number of responses by that number. Enter the result in the Total Score cell (zero has already been filled in where appropriate). Repeat the process for each column. When all the columns are complete, add the numbers appearing horizontally in the Total Score row and enter the result in the last cell of the table.

Table 1-5. System Category Survey (1 of 8)

I. System Platform	Factors				
a. Number of client/user organizations that will use the operational system 1 = One organization 2 = Two to three organizations 3 = Four or more organizations	1		2		3
b. Estimated number of concurrent end users of the system 1 = 1 2 = 2 to 10 3 = 11 to 20 4 = More than 21	1	2	3		4
c. Relative physical location of organizations that will use the system in production 1 = All at same location 2 = Within the same region 3 = In one or more regions 4 = All offices nationwide	1	2	3		4
d. Support software target platform 1 = Targeted for personal computers (PCs) 2 = Targeted for local area networks (LANs) 3 = Targeted for all platforms 4 = Targeted for mainframes	1	2	3		4
e. If the data is presently stored in an automated environment, what is the present location of the data? (Circle all that apply) 1 = One or more standalone PCs 2 = One or more LANs 3 = One or more mainframes	1	2			3
f. Estimated online data storage requirements 1 = Less than 100 MB 2 = 101 to 255 MB 3 = 256 MB to 1 gigabyte (GB) 4 = More than 1 GB	1	2		3	4
g. Estimated size of core random access memory (RAM) required by system 1 = 450 kilobytes (kB) or less 2 = 451 kB to 1 megabyte (MB) 3 = More than 1 MB	1			2	3
h. Estimated number of batch processing jobs per 24-hour period 1 = 0 2 = 1 to 2 3 = 3 to 5 4 = More than 5	1	2	3	4	

Return to Title Page

Table 1-5. System Category Survey (2 of 8)

I. System Platform	**Factors**				
i. Estimated average number of steps in each of the system's batch jobs 1 = N/A (no batch jobs) 2 = 1 to 15 3 = 16 to 30 4 = 31 to 50 5 = More than 50	1	2	3	4	5
j. Interfaces with office automation software 1 = System will need to interface with e-mail, spreadsheets, and word processing software 2 = Proposed system will not need to interface with office automation software	1				2
k. Text handling requirements 1 = System will be required to provide desktop publishing capabilities 2 = System will be required to provide full-screen text input and editing 3 = System will be required to provide only short or single-word text entry	1		2		3
l. Document search and retrieval functions 1 = Document search and retrieval functions will be requirements of the system 2 = Document search and retrieval functions will not be requirements of the system	1		2		
m. Estimated amount of offline historical or archived data generated per year 1 = 350 MB or less 2 = 351 to 700 MB 3 = 701 MB to 3.5 GB 4 = More than 3.5 GB	1	2		3	4
n. Estimated number of data elements in database 1 = Less than 200 2 = 201 to 400 3 = 401 to 600 4 = More than 600	1	2		3	4

Return to Title Page

Table 1-5. System Category Survey (3 of 8)

I. System Platform	Factors				
o. Complexity of algorithms and processing (Circle all that apply) 1 = Fewer than 50 simple mathematical calculations (add, subtract, multiply, and divide) will be performed by the system 2 = More than 50 simple mathematical calculations (add, subtract, multiply, and divide) will be performed by the system. 3 = 1 to 5 large, complex algorithms, central to the system, will be developed 4 = System will contain from 6 to 20 large, complex algorithms that will be used frequently 5 = Proposed system will contain more than 20 large, complex algorithms used frequently	1	2	3	4	5
p. Hardcopy reports 1 = System will produce no hardcopy reports 2 = System will produce fewer than 10 different hardcopy reports, generated only when requested by the user 3 = System will produce fewer than 10 reports automatically on a monthly or quarterly basis 4 = System will produce summary reports daily 5 = System will produce extensive, detailed reports daily	1	2	3	4	5
q. CASE tool usage 1 = Decision has already been made to use a CASE tool that targets PC environments 2 = Decision has been made not to use a CASE tool 3 = Decision has already been made to use a CASE tool that targets mainframe hardware	1		2		3
r. Interfaces to other systems 1 = System will interface only with PC- or LAN-based systems 2 = System will interface with LAN- and mainframe-based systems 3 = System will interface only with mainframe systems	1		2		3
s. Graphics 1 = System will be required to provide extensive graphic capabilities, including user-generated graphics and both ad hoc and predefined reporting shown graphically 2 = System will be required to generate simple predefined graphic reports 3 = System will not be required to provide graphic capabilities	1		2		3

Table 1-5. System Category Survey (4 of 8)

I. System Platform	**Factors**				
t. Data update and retrieval 1 = System will be required to provide real-time update and retrieval capabilities to online users 2 = System will provide online user interface to data repository that is updated every 1 to 4 hours 3 = System will provide online user interface to data repository that is updated daily 4 = System will provide online user interface to data repository that is updated every 2 or more days 5 = System will not provide an online interface for users	1	2	3	4	5
Response count Count the number of responses in each column and enter that count for each column in the appropriate cell to the right.					
Multipliers Use these multipliers to factor the weight for each column.	0	1	2	3	4
Total score For each column, multiply the response count by the multiplier and enter the product in the appropriate cell to the right.	0				
Add numbers in the row above and enter the total (between 0 and 80) in the cell to the right.					
II. Development Techniques	**Factors**				
u. Number of organizations (other than IT) that will provide input for requirements for the development effort 1 = One organization 2 = Two or three organizations 3 = More than three organizations	1		2		3
v. System will be developed by 1 = IT personnel only 2 = IT personnel and contractors 3 = IT and sponsor personnel 4 = IT and sponsor personnel, each with own contractors 5 = IT, sponsor, and end-user personnel, each with own contractors	1	2	3	4	5
w. Estimated elapsed time needed to develop system 1 = Less than 1 year 2 = 1 to 2 years 3 = 2 to 3 years 4 = 3 to 4 years 5 = More than 4 years	1	2	3	4	5

Table 1-5. System Category Survey (5 of 8)

II. Development Techniques	Factors				
x. System will support 1 = Known function that has been performed by HUD for years 2 = Known function or program that has recently changed slightly 3 = Known function or program that has recently undergone major changes 4 = Totally new program or function never before performed by HUD	1	2		3	4
y. System will cause 1 = No organizational changes 2 = Staff movements or additions 3 = Entire organizations to be disbanded or formed	1		2		3
z. System's functions 1 = Very similar to other systems previously developed and presently used at HUD 2 = Somewhat similar to other systems developed and used at HUD 3 = Similar to other systems used at HUD, but those systems were purchased or developed outside of HUD 4 = Do not resemble any existing systems at HUD	1	2		3	4
aa. Proposed system's programming language 1 = Is being used by HUD personnel to develop systems 2 = Has, for the most part, been used to develop systems. (A new version of the compiler will be used, but for the most part the language is unchanged.) 3 = Has never been used successfully to develop a system at HUD; however, staff or contractors have received training or have prior experience 4 = Is totally unknown to HUD staff and existing contractors, although there is a large base of knowledge in the software development community 5 = Is a leading-edge technology, with very few information systems professionals experienced in its use	1	2	3	4	5
bb. Proposed system's hardware and peripherals 1 = Are being used by HUD personnel 2 = Similar equipment has been or is being used at HUD 3 = Have never successfully been used to develop system at HUD; however, staff or contractors have received training or have prior experience 4 = Are totally unknown to HUD staff and existing contractors, although there is a large base of knowledge in the software development community 5 = Represent leading-edge technologies, with very few information systems professionals experienced in their use	1	2	3	4	5

Return to Title Page

Table 1-5. System Category Survey (6 of 8)

II. Development Techniques	Factors				
cc. Proposed system's support software 1 = Is being used by HUD personnel 2 = Similar software has been used in the past or is presently being used at HUD 3 = Has never successfully been used to develop a system at HUD; however, staff or contractors have received training or have prior experience 4 = Is totally unknown to HUD staff and existing contractors, although there is a large base of knowledge in the software development community 5 = Is a leading-edge technology, with very few information systems professionals experienced in its use	1	2	3	4	5
dd. Client and sponsoring organizations have expressed 1 = Very rigid systems requirements 2 = Flexibility with some requirements, but maintain a core of requirements that cannot be changed 3 = Desire to assemble team to determine requirements based on predetermined high-level specifications	1		2		3
ee. What is highest level of management with a direct interest in the system? 1 = Office Director(s) or Division Director(s) 2 = Assistant Secretary or Deputy Assistant Secretary 3 = Under Secretary or Deputy Under Secretary 4 = HUD Secretary		1	2	3	4
ff. Program area staff 1 = Most program area staff members are thoroughly knowledgeable about all phases of the program and functions that will be processed by the proposed system 2 = Most program area staff members are knowledgeable about a portion of the program and functions that affect their organization but have little knowledge of interaction of other organizations 3 = Most program area staff members have a thorough knowledge of their immediate function or portion of the program but do not know interaction with other members in their own organization 4 = Very few program area staff members understand how their actions impact the program or function; actions are based on instructions without an understanding of how those actions affect the overall program	1	2		3	4
gg. System can best be described as 1 = Transaction processing system 2 = Control and reporting system 3 = Decision support system	1		2		3

Table 1-5. System Category Survey (7 of 8)

II. Development Techniques	**Factors**				
hh. Overall system concept and requirements 1 = Generally well known and have much supporting documentation in the form of regulations, legislation, and existing procedures 2 = Generally well known, but no documentation exists or existing documentation is not up to date 3 = Generally well known, but some portions of requirements are vague, or conflicting understanding of requirements exists 4 = For the most part, not known or are vague	1		2	3	4
ii. Outcomes of implementing the system and the impacts to the agency 1 = Well understood and documented 2 = Generally understood, but all aspects not thoroughly known 3 = Not understood or known	1		2		3
jj. Tools (e.g., fourth-generation languages, CASE tools, simulation languages, data query languages, and screen formatters) to support alternate development techniques 1 = Not available for use by project development team 2 = Available for use, but members of development team have not used the particular tools available 3 = Available for use and project development team has been trained in their use 4 = Available for use and project team members have extensive experience in their use	1	2		3	4
Response Count Count the number of responses in each column and enter that count in the appropriate cell to the right.					
Multipliers Use these multipliers to factor the weight for each column.	0	1	2	3	4
For each column, multiply the response count by the multiplier and enter the product in the appropriate cell to the right.	0				
Add numbers in the row above and enter the total (between 0 and 60) in the cell to the right.					
III. Type of Effort	**Factors**				
kk. Development effort to be undertaken will cause 1 = No application design changes to existing system(s) 2 = Minor application design changes to reports or queries, input screens, and other input sources of existing system(s) 3 = Moderate application design changes to one subsystem and its associated input and output of existing system(s) 4 = Moderate application design changes to more than one subsystem and its associated input and output of existing system(s) 5 = Major application design changes throughout one or more systems	1	2	3	4	5

Return to Title Page

Table 1-5. System Category Survey (8 of 8)

III. Type of Effort	Factors				
ll. Development effort will cause 1 = Minor design changes to existing database(s); less than 10 percent of the database entities in any one database will be affected 2 = Moderate design changes to existing database(s); 10 to 40 percent of the data entities in any one database will be affected 3 = Major changes to existing database(s); affecting more than 40 percent of the data entities in any one database 4 = No design changes to existing database(s)	1		2	3	4
mm. Development effort is being performed to 1 = Correct errors detected during operation of an existing system or improve processing efficiency 2 = Respond to evolving or expanding user needs 3 = Respond to changes in the organization's environment	1			2	3
nn. The system 1 = Will replace an existing automated system 2 = Will replace an existing manual system 3 = Is totally new	1		2		3
oo. How old is the existing system? 1 = Less than 5 years old 2 = Between 5 and 10 years old 3 = Between 10 and 15 years old 4 = Between 15 and 20 years old 5 = More than 20 years old	1	2	3	4	5
pp. Estimated number of staff hours to complete the effort 1 = Less than 5,000 2 = 5001 to 12,500 3 = 12,501 to 25,000 4 = 25,001 to 50,000 5 = More than 50,000	1	2	3	4	5
Response Count Count the number of responses in each column and enter that count in the appropriate cell to the right.					
Multipliers Use these multipliers to factor the weight for each column.	0	1	2	3	4
For each column, multiply the response count by the multiplier and enter the product in the appropriate cell to the right.	0				
Add numbers in the row above and enter the total (between 0 and 24) in the cell to the right.					

1.3.2 Make Selection

The scoring from the system category survey factors and other decision criteria deemed critical by project management are used to assist in making the following determinations:

- Target platform of the proposed system
- Use of analysis, design, or system development or interactive development techniques (e.g., traditional prototyping, RAD, JAD)
- System classification (i.e., new development or an enhancement to an existing system)
- Use of COTS software, with or without modifications
- Tool usage, including CASE or automated tools
- Size and complexity of effort
- Whether the system is a major or minor effort
- A decision on the level of applicability and general guidance on the portions of the SDM that must be followed

Determinations made during this activity will be revisited at each SDM milestone. Information gathered as part of the project development effort may cause modifications to decisions made at this point in the effort. The modifications will be recorded when updates to the project documents are made.

1.3.3 Document Selection

The decisions made during these activities are recorded as part of the Feasibility Study that is prepared in accordance with HUD SDM documentation standards (Handbook 2400.15). Along with the recorded decisions, the completed system criteria survey provides a basis for describing the proposed system alternatives, method used to determine the category of the system, and the information supporting the category selection.

1.4 ASSESS FEASIBILITY OF DEVELOPMENT APPROACH

After the user’s needs have been identified and documented in a Needs Statement or ARN and the system category has been determined and documented, it is necessary to evaluate the various options available for meeting that need. Activities include the following:

1.4.1 Examine system objectives.

1.4.2 Evaluate alternatives.

1.4.3 Identify preferred approach.

1.4.4 Develop detailed feasibility study.

During this analysis, the objectives of the system are defined based on the needed functions described in the Needs Statement or ARN. Included in these system objectives are the high-level functional and performance objectives and any assumptions and constraints.

When the system objectives have been identified, the various alternatives for satisfying those objectives are determined. For each alternative, the costs in time and resources are estimated. A determination is then made as to the most feasible development alternative. Table 1-6 defines roles and responsibilities of key personnel for assessing feasibility of the project's development approach.

Table 1-6. Roles and Responsibilities for Assessing Feasibility of Development Approach

Role	Responsibility
Project Manager	Manages feasibility study and coordinates results of feasibility study with project sponsor.
User	Conducts feasibility study.
Developer	Provides input to feasibility study.
Computer Services	Provides input to feasibility study.
Quality Assurance	Reviews feasibility study for adherence to standards.

1.4.1 Examine System Objectives

The initial activity involved in determining feasible alternatives is to generate and examine system objectives by considering the functions defined in the Needs Statement or ARN. These objectives will form the basis for generating alternatives to meet the user's need.

Determine Proposed Functional Objectives

Analyze the anticipated functions of the system, considering such areas as new services, increased capacity, legislative and policy requirements, privacy and security requirements, and audit controls.

Identify Major Performance Objectives

Identify major performance objectives, considering such areas as reduced staff and equipment costs, increased processing speed, increased productivity, improved management information services, improved controls over automated decision-making system(s), and compliance with regulations.

Identify Assumptions and Constraints

Determine the assumptions and constraints, such as operational life of the proposed system; period of time for comparison of system alternatives; input, output and processing requirements; financial constraints; changing hardware, software, and operating environment; and availability of information and resources.

1.4.2 Evaluate Alternatives

Evaluate the development alternatives based on the system objectives. Document the alternatives, along with their associated costs and the criteria for evaluating the alternatives.

Determine Criteria for Evaluating Alternatives

Identify the criteria applicable to the development process that will be used to determine the most attractive system option. Such criteria typically include cost, priority, development time, ease of system use, or any combination.

Identify Alternatives To Be Considered

Identify and summarize each system alternative considered during the feasibility study, including use of one or more existing system(s), development of one or more new system(s), and the potential of purchasing an off-the-shelf system.

Outline Time and Resource Costs

For each system alternative identified, outline the time and resource costs, including the time and funding required for all activities of the lifecycle from definition through operation. It is imperative to use realistic estimates. When making time estimates, remember to include such factors as the current workload of personnel, staff absences due to vacation and illness, lead time for procurement of equipment and software, and staff training.

1.4.3 Identify Preferred Approach

Weigh each alternative identified during the evaluate alternatives process against the evaluation criteria and determine the most feasible development alternative.

1.4.4 Develop Detailed Feasibility Study

Describe and document the detailed feasibility study, describing the methodology and criteria used to determine the feasibility and the preferred approaches selected for fulfilling the system need.

Describe System Objectives

Document the functional and performance objectives of the system. Include such information as system output; system input; file descriptions; validation criteria; security, privacy, and control requirements; data storage and retrieval; and any interfacing system(s).

Describe Current Functional Procedures

Describe current functional procedures of any existing system, whether automated or manual. Specifically, document the major processing and data flow of the current system(s), volume of work currently processed, costs incurred in operating the current system, skill categories and number of staff required to operate and maintain the current system, equipment used by the existing system, and any other factors that are unique to the current system(s).

Describe Proposed System

Describe the proposed system, including such information as the overall system concept; improvements anticipated after successful implementation of the proposed system; anticipated impacts in the areas of equipment, software, personnel, operations, privacy, and security; building or office modifications; and projected costs of development and operations.

Document Results

Document the results of the analyses and studies related to the feasibility of the various system options in accordance with HUD SDM documentation standards (Handbook 2400.15).

Return to Title Page

1.5 PERFORM BENEFIT/COST ANALYSIS

Each alternative generated during the feasibility study is examined and a comparison is done on the benefits and costs of each. The current system, proposed system, and each alternative system identified in the feasibility study are described and their associated benefits and costs determined. These benefits and costs include developmental as well as operational (both one-time and recurring) costs. Activities for performing benefit/cost analysis include the following:

1.5.1 Identify the alternatives for development and operation.
1.5.2 Determine the costs per alternative.
1.5.3 Determine the benefits per alternative.
1.5.4 Develop a detailed benefit/cost analysis.

Table 1-7 defines roles and responsibilities of key personnel for performing the project's benefit/cost analysis.

Table 1-7. Roles and Responsibilities for Performing Benefit/Cost Analysis

Role	Responsibility
Project Manager	Manages benefit/cost analysis and coordinates results of benefit/cost analysis with the project sponsor.
User	Conducts benefit/cost analysis.
Developer	Provides input to benefit/cost analysis.
Computer Services	Provides input to benefit/cost analysis.
Quality Assurance	Reviews benefit/cost analysis for adherence to standards.

1.5.1 Identify Alternatives for Development and Operation

Identify the alternative approaches for the development and operation of the system, as determined in the feasibility study, and provide a brief description of each. In addition, provide a description of the current system (if one exists).

Describe Technical and Operational Characteristics of Current System

Briefly describe the current system, if one exists, by summarizing the functions of the system, identifying the hardware used, and identifying the input to and output from the system.

Describe Proposed System

Briefly describe the proposed system. Summarize the functional objectives of the system, and briefly describe the expected input to and output from the system.

Describe Alternative System

Briefly describe each alternative system discussed in the feasibility study. Include an outline for the basic design and identify any technical or operational characteristics that are specific to the system. In each description, address how the alternative did or did not meet the feasibility criteria.

1.5.2 Determine Costs per Alternative

Calculate all costs to develop and operate each alternative described in the feasibility study, including both one-time and recurring costs. Multiply costs to be incurred in the future by a present value factor, an inflation index that defines future costs in present dollars. This index takes into account items such as vendor and potential salary increases. This calculation brings the cost into present dollars.

Determine Development Costs

For each alternative system described in the feasibility study, estimate the cost of the Define, Design, and Build System phases. When determining the overall cost of development, include costs for personnel, equipment, developer training, and development tools.

Determine Operational Costs

For each alternative system described in the feasibility study, estimate the installation, operation, and maintenance costs of the system. Include such costs as personnel, equipment, operational site upgrades or changes, and staff training.

Determine Nonrecurring Costs

Determine the one-time costs that may be incurred with each alternative system described in the feasibility study. Include the costs of running a parallel system during system release, the potential disruption to existing business operations, etc.

Determine Recurring Costs

Determine the ongoing costs that will continue throughout the development and operation of each system alternative discussed. Include the costs required for operating and maintaining each alternative over the system’s life (e.g., equipment and software [lease, rental, and maintenance], data communications [lease and rental], travel and training, supplies and utilities, contractual support services, overhead).

1.5.3 Determine Benefits per Alternative

Evaluate each alternative for both the one-time and recurring economic benefits of developing and operating the system. Calculate in present dollars all benefits that will generate future economic returns. Also identify other benefits that cannot be quantified in dollar amounts.

Determine Nonrecurring Benefits

Determine nonrecurring benefits for each system alternative considered. Describe benefits that can be assigned dollar values. Include cost reductions resulting from improved system operations, such as reduction of resource requirements; improved operating efficiency; and improved data entry, storage, and retrieval. Also include benefits that enhance the value of the application system, such as improved resources utilization and reduced error rates.

Determine Recurring Benefits

Determine the recurring benefits of operating and maintaining each system alternative. Include equipment lease, rental, and maintenance; software lease, rental, and maintenance; data communications lease, rental, and maintenance; supplies and utilities; security and privacy; and travel and training.

Determine Nonquantitative Benefits

Describe the benefits that cannot be quantified in terms of direct dollar values, such as improved service, reduced risk of incorrect processing, improved information handling, and enhanced organizational image.

1.5.4 Develop Detailed Benefit/Cost Analysis

Describe and document the analysis, including the costing methodology used for performing the analysis. Document each alternative and its associated costs and benefits. Include a recommendation as to the most cost-beneficial alternative system, and provide the necessary information to support that recommendation.

Summarize Procedures for Conducting Analysis

Document the steps followed when performing the benefit/cost analysis. Include any reference material used, as well as charts and graphs if applicable.

Describe Techniques Used in Estimating and Computing Costs

Document how estimates and costs were derived. Include any reference material used, as well as charts and graphs if applicable.

Describe Benefits and Costs per Alternative

For each of the system alternatives analyzed, describe the associated costs and anticipated benefits that may be derived from developing and operating the system. Describe the costs and benefits in terms of recurring, nonrecurring, and nonquantitative.

Describe Most Advantageous Alternative

Based on comparative analysis of the benefits and costs, describe the alternative determined to be the most advantageous of the system alternatives described during the benefit/cost analysis.

Document Results

Document the results of the analyses and studies related to the costs and benefits of the various system options. Document the Benefit/Cost Analysis study in accordance with the HUD, Benefit/Cost Analysis Methodology : Volume I - Methodology and Volume II - Workbook, September 1995.

1.6 PERFORM RISK ANALYSIS FOR SYSTEM SECURITY

The ultimate purpose of risk analysis for system security is to help select cost-effective safeguards that will reduce security risks to an acceptable level. To accomplish this, an analysis is performed to estimate the potential losses that may result from system vulnerabilities and the damage from the occurrence of threats. Critical assets that must be protected are identified with consideration given to the environment in which these assets are stored and processed. One or more countermeasure(s) should be developed for each potential threat and vulnerability. Activities for performing risk analysis include the following:

1.6.1 Determine the type of risk assessment to be done.

1.6.2 Determine security requirements and/or specifications.

1.6.3 Determine baseline security requirements.

1.6.4 Determine threats and vulnerabilities.

1.6.5 Determine countermeasures.

1.6.6 Document detailed risk analysis.

Table 1-8 defines roles and responsibilities of key personnel for performing the project's risk analysis.

Table 1-8. Roles and Responsibilities for Performing Risk Analysis

Role	Responsibility
Project Manager	Coordinates results of risk analysis with the project sponsor.
User	Conducts risk analysis.
Developer	Provides input to risk analysis.
Computer Services	Provides input to risk analysis.
ADP Security	Reviews and approves risk analysis.
Quality Assurance	Reviews risk analysis document for adherence to standards.

1.6.1 Determine Type of Risk Assessment To Be Done

The determination of the type of risk assessment to be performed relates to the decision made during the determine category process (i.e., whether the proposed project will be a new development effort or an enhancement). The level of effort required to perform a risk analysis will be much greater for a new development effort than for an enhancement project.

Determine Whether Risk Assessment is of Modified or New System

Analyze the Needs Statement and the results of the feasibility study, as well as the determination made during the categorization process, to ascertain whether the solution to the information management need will result in a new development effort or in the maintenance or enhancement of an existing system.

Determine System's Place in System Development Lifecycle

If the development effort is determined to be maintenance or enhancement of a system currently in development or production, identify the system's location in the system development lifecycle. Describe the location of the system in terms of its phase of development or the length of time it has been in operation.

1.6.2 Determine System Security Requirements and Specifications

Based on the environment, scope, sensitivity of the data, and criticality of the proposed system to the project sponsor and users, assess the security requirements and specifications necessary to safeguard the system and its corresponding data. Include such information as privacy requirements, estimated dollar value of assets, and contingency planning requirements.

1.6.3 Determine Baseline Security Requirements

Analyze the processes and procedures required of the new system or the system to be replaced and the sensitivity of the data the system will be processing to determine inherent security risks. Determine the security controls that will be required to adequately counteract these security risks.

Determine Sensitivity Level of Data Being Processed

Evaluate the data being processed to determine whether the level of sensitivity requires safeguards, such as the application of security controls. In this evaluation, include input and output data, internally processed data, and data transmitted to or from the system.

Determine User Security Investigation Level and Access Need

Analyze the system's end users, including those having direct access to the system and those who will indirectly receive output from the system. Determine the levels of security investigation and system access required for each user.

1.6.4 Determine Threats and Vulnerabilities

Evaluate the proposed system and its operational environment for potential threats and vulnerabilities. Estimate the potential security risks in the following areas:

- Physical: Determine the vulnerability of the computer room and the impact of environmental hazards on the computer, related equipment, and their contents, including the following considerations:
  - Location of computer complex
  - Construction of computer room
  - Control of access
  - Environmental protection
  - Security of media library
  - Security of microcomputers
  - Housekeeping in computer room
  - Management of waste
- Communication: Evaluate the system for threats to the privacy and authenticity of telecommunications, including the following:
  - Password system
  - Sign-on process
  - Encryption procedures
  - Dial-up lines

  - LANs

- Hardware: Review the system's current or proposed hardware configuration, including the following:
  - Mainframe computer security
  - Operator's console
  - Mini- and microcomputer security
  - Remote terminals
  - Security of system peripherals (e.g., printers, disk drives, tape drives)
  - Security of laptop computers and other portable equipment
- Software: Review the system software for security risks and potential vulnerabilities, including the following:
  - Features of the operating system
  - Functions of the applications
  - Database security
  - Control of related documentation
- Other: Give special attention to the following when evaluating for potential risks and vulnerabilities:
  - Frequency of periodic risk assessments of the operational system
  - Identification of organizations responsible for managing identified risks and maintaining countermeasures
  - Development of contingency plans

1.6.5 Determine Countermeasures

Analyze the identified security threats and potential vulnerabilities of the proposed system, and determine the necessary measures to be taken to safeguard this system. Evaluate the identified measures for appropriateness and cost efficiency, and formulate a recommendation identifying those measures deemed suitable for implementation.

Review Impact of Potential Hazards

Evaluate the potential threats to, and vulnerabilities of, the proposed system to determine the impact they may have on the system. For each potential risk or vulnerability identified, provide an assessment of the magnitude of that impact in the event of an occurrence.

Determine Availability of Potential Safeguards

Research the security-related technology currently available and that which is projected to be available at the time the system is scheduled for operation. Determine potential safeguards against all identified risks and vulnerabilities, including those safeguards that may already be in place at HUD.

Return to Title Page

Determine Safeguards Appropriate for Risk Environment

Review each of the identified safeguards and determine whether it is appropriate for use within the system's operational environment. Indicate its level of compatibility with the guidelines for operating information systems at HUD.

Determine Lifecycle Costs of Each Operationally Acceptable Safeguard

Estimate the cost to develop, install, and operate each of the proposed system safeguards. Include user training and maintenance, if required, in these estimates.

Estimate Effect of Safeguard on Each Identified Risk

For each of the proposed system's identified risks and vulnerabilities, estimate the extent to which the recommended safeguard will be effective in preventing or minimizing that threat or vulnerability.

Determine Economic Feasibility

Contrast the lifecycle costs of each of the potential safeguards against the financial impact of the security risks they are designed to prevent. Consider the effect each safeguard is projected to have on minimizing those security risks. Determine whether the benefits achieved by these safeguards outweigh their operational and developmental costs.

Prepare Risk Reduction Decision Study

Prepare a study outlining the potential security risks to the system to be developed or replaced. Provide a detailed description of the security safeguards that are being recommended to counteract those risks. Include an analysis of the safeguards supporting the feasibility of their implementation.

1.6.6 Document Detailed Risk Analysis

Document the studies that were performed to determine the potential risks and vulnerabilities of the proposed system and the recommended safeguards to counteract those risks and vulnerabilities.

Describe Threats and Vulnerabilities

Provide a description of each of the threats and vulnerabilities to the existing or proposed system that were uncovered in the earlier analysis process. List the described threats and vulnerabilities in the order of the magnitude of their impact on the system's processes.

Describe Countermeasures

List each of the proposed system safeguards and explain its use as a countermeasure to the identified threats and vulnerabilities of the system. Indicate to what extent the countermeasure is anticipated to safeguard the system.

Describe Security Requirements

List the security requirements imposed on the system by HUD Automated Data Processing (ADP) security guidelines and indicate the ability of the proposed system safeguards to meet these requirements.

Prepare Risk Analysis Document

Develop the Risk Analysis Document in accordance with HUD SDM documentation standards (Handbook 2400.15) and project guidelines.

Describe the risk analysis study that was performed and the system risks and vulnerabilities that were uncovered. Include a description of the proposed countermeasures to these security risks, and describe how these countermeasures will safeguard the system from the these risks.

1.7 RECORD SYSTEM DECISIONS

The decisions made during development of the Initiate Project phase documents (e.g., Needs Statement, risk analysis, feasibility study, benefit/cost analysis) are recorded in an SDP. The SDP is submitted to the appropriate review board for approval or disapproval. Based on the SDP and supporting analysis, the appropriate review board makes the decision to continue development of the project. This decision is also recorded in the SDP, as is the selected alternative if the decision is made to continue with the development effort. Activities for recording system decisions include the following:

- 1.7.1 Summarize issues.
- 1.7.2 Outline accomplished and projected schedules.
- 1.7.3 Develop a System Decision Paper.

Table 1-9 defines roles and responsibilities of key personnel for recording the project's system decision.

Table 1-9. Roles and Responsibilities for Recording System Decisions

Role	Responsibility
Project Manager	Manages development of SDP.
User	Develops SDP.
Quality Assurance	Reviews SDP for adherence to standards.

1.7.1 Summarize Issues

Summarize for management the information gathered from the previously produced documents. If management, through the approval process, decides to proceed with the development of the system, determine the criticality of the proposed system. The criticality is determined through comparisons with other potential systems identified in the HUD Department ISP. Include the following in the SDP:

- Identify the information management need as detailed in the Needs Statement or ARN.
- Summarize the current system procedures as detailed in the feasibility study.

- Indicate the anticipated benefits of the new system as detailed in the feasibility study.
- Determine the importance of the proposed project to HUD as documented in the risk analysis.
- Summarize, by fiscal year, the cost to develop and operate the system as computed in the benefit/cost analysis.
- Identify the funding source (the organization responsible for most, if not all, of the cost to develop and operate the system; typically, the project sponsor or primary user).

1.7.2 Outline Accomplished and Projected Schedules

Document the schedule of activities that transpired during the system development decision processes and include them in the SDP. Provide a projected schedule of activities and milestones by phase for the full lifecycle of the system. Include the actual or projected beginning and ending dates for each phase, as well as a running total of the actual or projected development and operation costs broken out by phase and totaled by fiscal year.

1.7.3 Develop System Decision Paper

Using the summary information and projected schedules, develop the SDP to document and support management's decision on resolving the need.

Describe Summarized Issues

Provide a summary of the progress of the proposed project to date and incorporate it into the SDP. Include any milestones met or impediments to progress that were encountered.

Describe Accomplished and Projected Schedules

Prepare a schedule detailing completed and anticipated project activities, including the milestones.

Prepare Document

Prepare the SDP in accordance with HUD SDM documentation standards (Handbook 2400.15) and project guidelines. Include a summary of the issues encountered, the decisions reached, and the accomplished and anticipated project schedules, with accompanying actual and projected resource expenditures.

1.8 REFINE PROJECT PLAN

The Project Manager controls the project by monitoring phase activities; taking corrective action where necessary; refining the project plan to account for changes due to actions taken in the current phase and new information for upcoming phases; and reviewing the planned quality process developed by users, developers, testers, and QA for the next phase. Activities include the following:

1.8.1 Update Project Plan.

1.8.2 Review planned quality process.

Return to Title Page

Table 1-10 defines roles and responsibilities of key personnel for updating the Project Plan.

1.8.1 Update Project Plan

Update the Project Plan with cost, schedule, and budget data for the current phase (Initiate Project) to the level of detail necessary to reflect the status of the project.

Review the plan and determine whether any activities for subsequent phases are affected by the completion status of the current phase activities.

Adjust schedules and resource requirements for activities in the following phases, if necessary, and assign starting and ending dates for activities that have been affected. Take into account that the starting date for some activities may be dependent on the completion of other activities.

Update milestones, schedules, resource requirements, and programmatic risk management strategy for the remainder of the project.

Table 1-10. Roles and Responsibilities for Refining Project Plan

Role	Responsibility
Project Sponsor	Approves changes to the project schedule. Approves changes to the Project Plan.
Project Manager	Updates the Project Plan to include actual costs incurred and specific activities accomplished for the Initiate Project phase; and revises project schedule, plans, strategies, resources, and requirements for the subsequent phases, as required. Reviews, for approval, the quality process planned by users and developers for the project phases.
User	Provides input to project manager on required Define System phase tasks and activities. Participates in the Project Plan review. Works with developer to identify and plan the quality process for the project phases.
Developer	Provides input to project manager on required Define System phase tasks and activities. Participates in the Project Plan review. Works with user to identify and plan the quality process for the project phases.

1.8.2 Review Planned Quality Process

Users, developers, and QA work closely together to determine the process that will be used to build the product with the desired quality. During the Initiate Project phase, activities include identifying and planning the quality process for the project phases and identifying the standards and procedures to be used to develop the product. Near the end of each project phase, the planned quality process is reviewed and updated for the next phase. The project manager reviews the quality process for approval and execution.

Return to Title Page

1.9 REVIEW AND APPROVE DOCUMENTS AND DELIVERABLES

The review and approval of documents and deliverables is an iterative process as each phase provides more definition and products are revised. Additionally, the approval process offers senior management insight into the project. Activities include the following:

1.9.1 Conduct review for Initiate Project phase documents and deliverables.

1.9.2 Obtain approval of project documents and deliverables.

Table 1-11 defines the roles and responsibilities of key personnel for review and approval of deliverables and documents.

1.9.1 Conduct Review for Initiate Project Phase Documents and Deliverables

Conduct a project review with project personnel and system stakeholders to ensure that the project documents and deliverables for the Initiate Project phase include the necessary level of detail to evaluate a proposed solution to the user's information management need and meet the appropriate HUD and project standards and guidelines. Use the appropriate document review checklist in Appendix E for Initiate Project phase deliverables to aid the document reviews.

Table 1-11. Roles and Responsibilities for Review and Approval of Deliverables and Documents

Role	Responsibility
Project Manager	Determines required review level and schedules reviews. Attends reviews and presents deliverables. Follows up on reviewer recommendations. Obtains concurrence and approvals for deliverables and associated management summaries.
User	Assists in preparation of deliverable reviews. Attends deliverable reviews.
Developer	Assists in preparation of deliverable documents. Provides technical expertise during review process.
ADP Security	Reviews phase products to ensure security requirements have been incorporated.
Project Sponsor	Participates in deliverable reviews.
Configuration Management	Assists project manager with review preparation.
Computer Services	Participates in review of documents and deliverables.
Quality Assurance	Ensures all review procedures are followed, as required. Reviews products to ensure they meet all applicable HUD standards and guidelines.

A minimum of 10 working days before the scheduled review, notify the personnel required to attend and provide each with a copy of the product for their pre-review. Discuss all comments or objections that are raised during the review, and reach consensus on one of the following before the review session terminates:

- The document is correct and complete, as is, without any further changes

- Additional changes that need to be made are minor and do not require further review. In this case, the updates should be made and change pages should be distributed by an agreed-upon date.
- Required changes will have a major impact on the plan. A second review must be scheduled. The changes must be incorporated and the resulting change pages distributed a minimum of 10 working days before the second review.

Prepare a management summary for each document that includes the essential data collected in the document, lists conclusions that may be drawn from the document, and describes potential impacts on the project. Submit documents to the appropriate review board for approval. Documents for approval include the following:

- Needs Statement or ARN
- System Decision Paper
- Feasibility Study
- Benefit/Cost Analysis
- Risk Analysis
- Project Plan (updated)

1.9.2 Obtain Approval of Project Documents and Deliverables

Present the project documents and deliverables to the chairperson of the appropriate review board for approval at least 10 days before the scheduled decision date. Include the management summary information, approval (sign-off) record, the System Decision Paper, and any other required or requested information. The review board chairperson coordinates review board comments, recommendations, and approval signature(s) and returns the approval record to the project manager. Recommendations on the approval record are addressed by the project and the document should be resubmitted to the board if requested. Approval will be assumed if there is no response, or the response is “no comment.” Project approval records are maintained by the project’s configuration management (CM) function. A copy is inserted into the central project library.

The project proceeds to the next phase after all project documents and deliverables are approved.

1.10 PERFORM CHANGE CONTROL ACTIVITIES

A CM function is created to include a project Change Control Board (CCB), a configuration status accounting (CSA) function, and a central project library. All products developed during the project’s development phases are baselined and subjected to change and version control. Baselined items are assigned new version control numbers as they undergo changes (e.g., an update or rewrite).

Configuration management reports are provided to the project manager as requested or required.

Table 1-12 defines the roles and responsibilities of key personnel for performing configuration change control activities for approved baselined products.

Return to Title Page

Table 1-12. Roles and Responsibilities for Performing Change Control Activities

Role	Responsibility
Project Manager	Approves controlled products for distribution. Approves Configuration Management Plan and supporting activities.
Configuration Management	Establishes Configuration Control Board (CCB). Develops configuration status accounting (CSA) function. Establishes a central project library. Establishes change control for new products. Performs change control for new products. Provides status reports to project management as requested or required.
Quality Assurance	Audits products to ensure only approved products are baselined. Validates establishment of project library.

The configuration management function prepares the baseline for configuration control. The baseline includes any new technical and document deliverables that will compose the project configuration baseline. For the baseline items, configuration management performs the following activities:

- ***Develop Configuration Management Plan (CMP).*** If desired, create the CMP as a subsection of the Project Plan or as a standalone document. The CMP provides the plan for recording change processing and implementation status and identification and documentation of inconsistencies among successive baselines.

- ***Establish project CCB.*** The project sponsor establishes and heads the Change Control Board (CCB) for the project. The project CCB establishes the control mechanism for the program office that has identified the need for which the project has been initiated. The goal of the project CCB is to baseline system products and review for approval (and implementation) any changes to the baseline. Criteria for approval will include cost and schedule impacts of proposed changes. The project sponsor will appoint board members from the following organizational areas:

  – Program officials

  – System users

  – External stakeholders, such as Federal oversight agencies

  – IT representative(s), such as project leader, project QA, and technical support

  As head of the CCB, the project sponsor does initial reviews of requirement changes to determine if they are changes in scope or clarifications or derivatives of existing baselined requirements. The project sponsor holds CCB meetings to review and approve changes of scope and authorizes the project team to implement clarifications and derivatives.

- ***Establish central project library*.** Establish a library for baselined documents and deliverables. The library, controlled by the project CM function, provides a single, up-to-date repository for the project's documentation and software products.
- ***Establish configuration status accounting reporting*.** Record and monitor changes to controlled documentation and software. Provide CSA reports to appropriate personnel to furnish up-to-date status of approved, baselined products.
- ***Establish change control and version number procedures*.** Identify procedures needed to ensure that changes are accomplished in an organized manner with absolute traceability and accountability and to ensure proper implementation of project CM requirements. For the version number, follow procedures established by HUD. The version numbering enables the project's CM to monitor updates to the product and assist in its distribution. For revised baselined products, review changes made to products to ensure that the changes have been made as described in the supporting documentation.
- ***Assign version number*.** The version number format must follow conventions established by HUD. The version numbering enables the project's configuration management to monitor updates to the product and assist in its distribution. For revised baselined products, review changes made to products to ensure that the changes have been made as described in the supporting documentation.
- ***Store approved version in central library.*** After each baseline product has been completed and approved according to HUD procedures, store the approved version in the project's central library. The project's CM function controls access to the library.
- ***Record product information in inventory log*.** Maintain an inventory log that includes the title of the product, release date, version number, name and version or model numbers of the software and hardware used in the development of the product, name of the organization responsible for development of the product (usually the sponsoring organization), and the product distribution list.
- ***Distribute copies of products as required*.** The project's configuration management function distributes copies of the product according to a distribution list maintained as part of the inventory log information, with distribution based on need and the security level of the product.
- ***Archive old version of products*.** Archive and retain outdated versions of all products for the required period of time, in accordance with HUD guidelines.

This page intentionally left blank.

Section 2. Define System

2.0 Define System

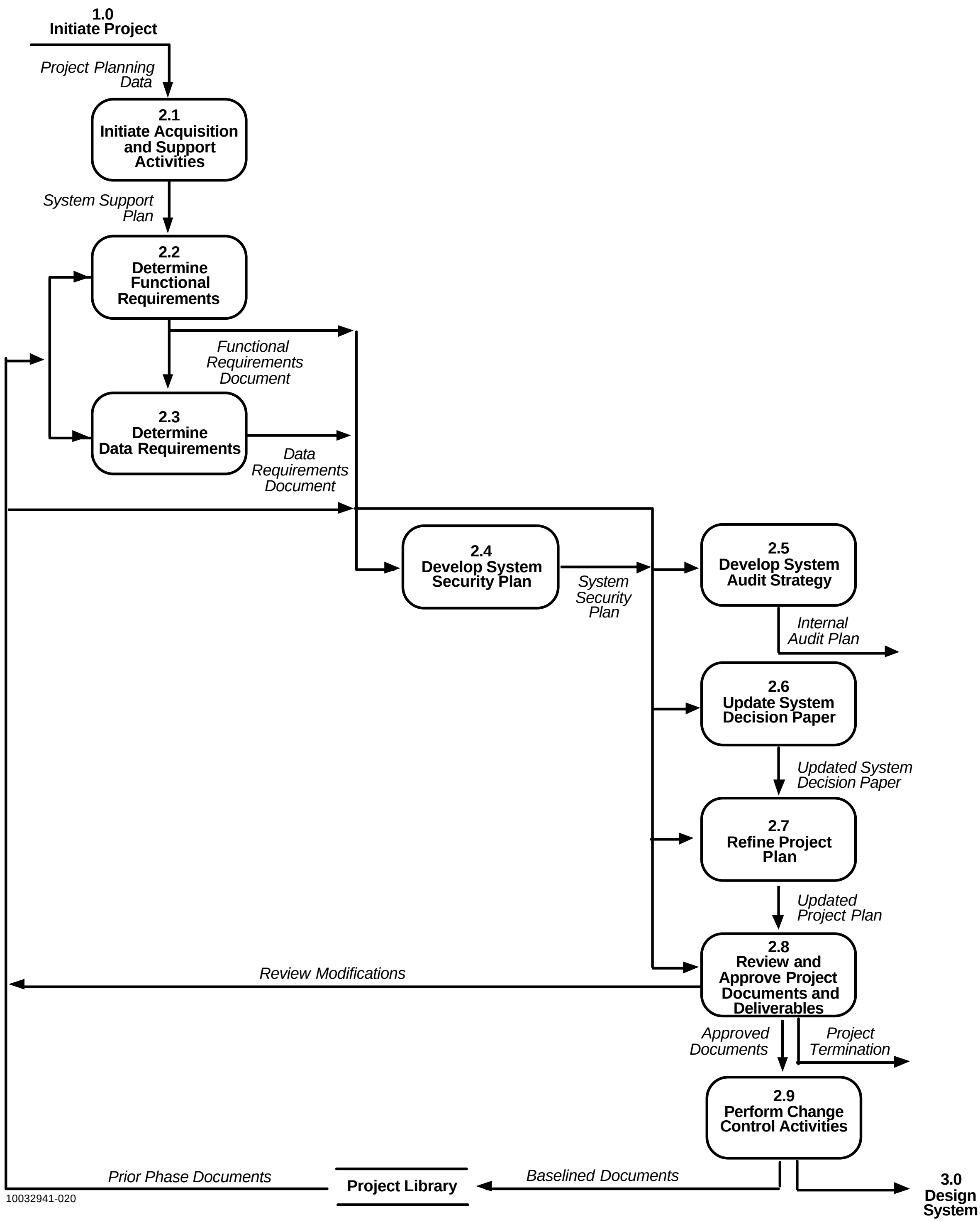

Figure 2-1. Process Flow for Define System Phase

Return to Title Page

2.0 DEFINE SYSTEM

Purpose

The Define System phase expands system objectives into specific, detailed functional and data requirements, which then form the basis for the detailed design of the system during the Design System phase. At the end of the Define System phase, a complete detailed description of the system functions is available to guide the design and subsequent activities of system development. Figure 2-1 highlights the process flow for the Define System phase.

Overview

During the Define System phase, the user's information management need is translated into a data processing solution. The users and developers work jointly to define the system's functional and data requirements, as follows:

- Emphasize what needs to be designed, not how to design it.
- Define and allocate requirements from the top down into subcomponents that are as independent as possible so that the system is easy to maintain, understand, and review in segments.
- Provide a conceptual or logical data model of the proposed system, independent of any data store, access method, or hardware.
- Document the requirements in a matrix that traces system objectives to functional and data requirements and to subsequent phase products that satisfy and validate the requirements.
- Submit requirements to walkthroughs (peer reviews) to discover potential product flaws and provide critical feedback.

The following activities are conducted during the Define System phase:

- Determine the scope of the system objectives and major functions.
- Begin hardware and software acquisition activities necessary for system development.
- Define the boundary of the existing system along with its inputs and outputs. Break down system objectives and major functions to processes, and processes to the single-function level. Write function definitions to describe the actions or activities taking place within the processes. Document definitions in the Functional Requirements Document.
- Develop a logical data model showing how the data within the system are related in order to minimize duplication of data, provide flexibility, and allow the data to be mapped to a wider variety of database designs. Define data attributes in the model. Document definitions in the Data Requirements Document.
- Cross-check functional and data requirements to ensure consistency among requirements.

- Use the requirements matrix to trace functional and data requirements to system objectives and needed functions approved by the Technology Investment Board (TIB).
- Determine data conversion and report requirements.
- Work with the ADP Security Office to develop a System Security Plan that describes the controls needed to counteract the risks determined in the Initiate Project phase.
- Maintain the Project Plan. Document project expenses for this phase and any new or revised estimates for subsequent phases to reflect cost and schedule for completing the project.
- Review the project deliverables listed in Table 2-1.
- Present the System Decision Paper and all required supporting documentation to the appropriate review board at the end of this phase for approval. Projects that receive review board approval will continue to the next phase of system development.

Table 2-1. Define System Phase Functions and Products

Define System Functions	Products
2.1 Initiate Acquisition and Support Activities	System Support Plan
2.2 Determine Functional Requirements	Functional Requirements Document
2.3 Determine Data Requirements	Data Requirements Document
2.4 Develop System Security Plan	System Security Plan
2.5 Develop System Audit Strategy	Internal Audit Plan
2.6 Update System Decision Paper	System Decision Paper (updated)
2.7 Refine Project Plan	Project Plan (updated)
2.8 Review and Approve Documents and Deliverables	New and revised products Management Summary[1]
2.9 Perform Change Control Activities	Change control records

[1] A management summary is prepared for each product produced or revised during the Define System phase. This one-page summary includes a summary of the essential data collected in a document product, conclusions that may be drawn from the document, and potential impacts on the project, if applicable.

Standards and Guidelines

Follow HUD SDM documentation standards (Handbook 2400.15) and project guidelines during development of Define System phase products.

Roles and Responsibilities

Throughout the Define System phase of development, key personnel are required to perform various activities. Table 2-2 lists types of personnel required and the activities for which they are responsible.

Table 2-2. Roles and Responsibilities for Define System Phase (1 of 3)

Role	Responsibility
Project Sponsor	Establishes coordination with appropriate internal and external organizations, and assists in maintaining communications with those organizations. Participates in reviews. Provides input to and approves functional and data requirements, changes to project schedule, and changes to Project Plan.
Project Manager	Initiates acquisition support activities. Determines resource and acquisition requirements. Updates System Decision Paper, as necessary. Prepares System Support Plan. Determines needed changes to prior phase documents, and ensures revision in accordance with the overall Project Plan. Coordinates review of Define System phase deliverables. Determines level of required review and schedules required reviews. Attends reviews and presents deliverables. Coordinates with Inspector General's staff. Ensures that Define System phase deliverables are updated to include any recommendations received during reviews. Obtains appropriate concurrence and approvals of updated documentation. Obtains appropriate concurrence and approval for Define System phase deliverables. Updates Project Plan to include actual costs incurred and specific activities accomplished for the Define System phase, and revises project schedule, plans, strategies, resources, and requirements for subsequent phases, as required. Reviews, for approval, the planned quality process identified by users and developers for the next phase. Works with Central Information Management to determine if there is any duplication of effort.
Project Development Team (User and Developer)	Confirms and supports resource requirements. Provides input to resource and acquisition requirements. Provides input to Project Manager on required Define System phase activities. Prepares Functional Requirements Document. Prepares Data Requirements Document.

Return to Title Page

Table 2-2. Roles and Responsibilities for Define System Phase (2 of 3)

Role	Responsibility
Project Development Team (User and Developer) (Cont'd)	Creates requirements matrix Prepares System Security Plan. Revises prior phase products when applicable. Assists with preparation of Define System phase deliverables for review. Attends deliverable reviews. Provides technical input to the review process, as required. Reviews and updates the quality process planned for the next phase.
Computer Services	Provides technical information and support during the acquisition process. Initiates preliminary capacity planning estimates. Participates in review of Functional Requirements Document. Participates in review of Data Requirements Document.
ADP Security	Provides input to functional and data requirements analysis (optional). Assists in development of System Security Plan. Reviews Define System phase deliverables to ensure that all necessary security requirements have been incorporated. Reviews revised products to ensure that the necessary security standards and guidelines have been addressed.
Inspector General (IG) Staff	Coordinates with project manager, as necessary. Develops the Internal Audit Plan to incorporate information received from the project manager. Original plan remains with IG and copy goes to central project file.
Quality Assurance	Reviews revised products to ensure they meet applicable HUD and project standards and guidelines. Reviews Define System phase deliverables to ensure they meet applicable HUD and project standards and guidelines. Ensures that all review procedures are followed, as required. Coaches users and developers in the planning for quality in the next phase.
Configuration Management	Provides input to the Project Plan. Performs version control for revised products. Provides Configuration Management (CM) reports to project management, as requested or required. Assists the project manager with preparation of Define System deliverables for review.

Table 2-2. Roles and Responsibilities for Define System Phase (3 of 3)

Role	Responsibility
Project Data Administrator	Supports generation of application requirements. Adds application-specific entities or objects to the application data model that was originally taken from the enterprise-wide data model. Generates and revises the problem domain data model. Reviews and reconciles data models, and ensures that data elements conform to HUD naming standards. Participates in review of Data Requirements Document.

2.1 INITIATE ACQUISITION AND SUPPORT ACTIVITIES

Begin activities related to acquisition of hardware, software, communications support, and any professional services required to support the project. Perform these activities in coordination with the effort for determining functional requirements. Depending on the scope of the project, such activities may include acquisition of operating system hardware; supporting software packages; services of the vendor(s) who will provide and maintain these tools; and contractor support for the development, testing, and maintenance of the system. Include the following activities:

2.1.1 Initiate system support activities.

2.1.2 Initiate acquisition activities.

2.1.3 Document system support plan.

Table 2-3 defines roles and responsibilities of key personnel for initiating acquisition and support activities.

Table 2-3. Roles and Responsibilities for Initiating Acquisition and Support Activities

Role	Responsibility
Project Sponsor	Establishes coordination with appropriate internal and external organizations. Assists in maintaining communication with those organizations involved in acquisition and support activities.
Project Manager	Initiates acquisition and system support activities. Determines resource and acquisition requirements. Prepares System Support Plan.
User	Provides input to acquisition requirements.
Developer	Identifies system support and acquisition requirements.
Computer Services	Provides technical information and support during the acquisition process.

2.1.1 Initiate System Support Activities

Ensure that supporting organizations, such as Computer Services, Operations, and Maintenance Group (CSOMG), Development Technology Assessment Division (DTAD), and the ADP security staff are aware of the project's requirements and currently have the necessary resources to meet those requirements (or are taking appropriate steps to acquire them). Coordinate with other projects being developed to determine if they have similar requirements for hardware or software; if so, steps should be taken to combine the resources of those projects to acquire the necessary hardware or software.

Coordinate with CSOMG and DTAD

Ensure that CSOMG and DTAD, as supporting organizations, are working concurrently with the project to meet its requirements. CSOMG and DTAD must be kept informed regarding the hardware and software requirements of the project, as well as changes to the implementation and processing schedule. Include the following activities:

- Inform CSOMG and DTAD of the system’s production requirements.
- Coordinate with CSOMG and DTAD to ensure that they have ample time to prepare for production involved in the project. Inform CSOMG and DTAD of such elements as performance requirements, amount and type of output expected, distribution and archiving of output, and number of expected users. Update CSOMG and DTAD on any changes made to production requirements so that they will be able to support the system when it is implemented.
- Inform CSOMG and DTAD of the system’s development schedule.
- Provide CSOMG and DTAD with an approved copy of the project’s schedule and ensure that they are made aware of any changes to it.

2.1.2 Initiate Acquisition Activities

Based on the decisions made for support requirements, estimate hardware and software requirements for development and operation of the system, and begin acquisition activities for each.

Initiate Software Acquisition Activities

Review and evaluate programming language(s), compilers, and supporting software packages (e.g., CASE and project planning software) to be used for the system. Follow procedures required to procure the chosen software.

Initiate Hardware Acquisition Activities

Begin the review, evaluation, and acquisition of HUD-approved hardware that will be needed for development of the system (e.g., workstations and printers).

2.1.3 Document System Support Plan

Document those activities that require assistance from support organizations as determined during the coordination process. Estimate a completion date for each activity, and identify the point of contact within each supporting organization. Identify hardware and software acquisition activities and document them in the System Support Plan in accordance with HUD SDM documentation standards (Handbook 2400.15). The System Support Plan may be included as a subplan under the Project Plan or may be developed as a standalone document. Provide the System Support Plan to the affected organizations for their review and concurrence. If any changes to the plan's activities or schedule are recommended, integrate those responses into the plan.

2.2 DETERMINE FUNCTIONAL REQUIREMENTS

Functional requirements provide a basis for mutual understanding between users and developers of the initial definition of the system and the environment in which it will operate. Users and developers work jointly to identify requirements. In situations where users are solely responsible for identifying functional and data requirements, hold periodic walkthroughs during the identification process so that users can present requirements to developers and receive feedback as to the clarity and completeness of the requirements. Other information determined during the identification of functional requirements includes control objectives, cost constraints, and performance requirements. Activities include the following:

2.2.1 Identify functional requirements.

2.2.2 Define performance requirements.

2.2.3 Define operational environment.

2.2.4 Identify control objectives.

2.2.5 Identify cost constraints.

2.2.6 Document functional requirements.

Table 2-4 defines roles and responsibilities of key personnel for determining functional requirements.

Table 2-4. Roles and Responsibilities for Determining Functional Requirements

Role	Responsibility
Project Sponsor	Provides input and approves functional requirements. Assists in determining resource requirements.
Project Manager	Determines resource requirements.
User	Works jointly with developer to determine functional requirements and dependencies.
Developer	Works jointly with the user to determine functional requirements and dependencies. Prepares Functional Requirements Document.
Computer Services	Participates in review of Functional Requirements Document. Initiates preliminary capacity planning estimates.
ADP Security	Provides input to functional requirements identification.
Quality Assurance	Monitors and reviews functional requirements activities and products.

2.2.1 Identify Functional Requirements

Identify all functional requirements the system is expected to fulfill. These requirements must be associated with approved TIB activities and be within the scope of Initiate Project phase documents. Describe the current procedures for the system, if applicable, and the procedures for the proposed system. Identify organizational, operational, and user impacts. Determine assumptions and constraints.

Define Current Procedures

Briefly describe current methods and procedures employed to satisfy existing information requirements. Provide one or more graphic representations that depict existing data flow through the functional system, from data acquisition through its processing and eventual output. Complement the graphic(s) with an explanation of the sequence of operational functions performed by the system for decision-making. In addition, at a minimum, include the following information:

- Organizational and personnel responsibilities
- Equipment being used
- Input and output, including volume, sources, and frequency
- Provisions in the existing system design for operation in degraded modes or at alternate sites in the event of emergency, disaster, or accident
- Deficiencies, such as time delays

Define Proposed Procedures

Provide an explanation of how the proposed system will interact with functional processes that the proposed system will support. Identify products from other systems that will be used with, or become part of, the proposed system.

Provide a qualitative and quantitative summary of the benefits of the proposed system. Explicitly identify the requirements to be satisfied by the proposed system. Include information flows, business logic, manual procedures, and security considerations. Give a comparison of deficiencies and identify any additional requirements, with appropriate explanations. When improvements to the existing methods and procedures are required, state the extent of the anticipated improvements.

Identify Organizational, Operational, and User Impacts

Describe any organizational impacts that may occur, including modification of responsibilities and addition or elimination of responsibilities that may be required when using the proposed system. Identify requirements for the number and skills of additional personnel, including changes in authorized strength, location, and position. Following an emergency, disaster, or accident, if functions of the proposed system need to be restored at one or more alternate sites, identify the number and skills of personnel needed for contingency operation at each site.

Include a discussion of operational impacts on the organization during use of the proposed system. In this discussion, consider the following:

- Proposed interface between user and primary or alternate computer operating centers
- Impact on user when changing from current operational procedures
- New data sources
- Quantity, type, and timeliness of data to be submitted for use in the system
- Data retention requirements
- Modes of user operation based on emergency, disaster, and accident

Discuss all user impacts on development and all user efforts required before implementation of the system, such as development or modification of data, systems testing, or required training.

Define Assumptions and Constraints

Describe any user assumptions and constraints that will affect development and operation of the system. Discuss any limitations affecting desired capability and identify any desired capabilities that will not be provided by the proposed system. Include any anticipated operational changes that will affect proposed operation of the system.

2.2.2 Define Performance Requirements

Identify all performance requirements the system is expected to meet. Include performance-related characteristics, such as online response time, batch turnaround time, capacity limits, accuracy and validity requirements, input and output effects on performance, and failure contingencies.

Determine Acceptable Online Response Time

Define the desired online response time and describe it in clock time, such as seconds. It may be a one-time representation of the maximum time allowed or a range of time, such as “from 1 to 5 seconds.”

Determine Acceptable Batch Turnaround Time

Define the desired batch response time, expressed in clock time, such as hours or days. This may be a one-time representation of the maximum time allowed or a range of time, such as “from 1 to 3 working days.”

Determine Capacity Limits

Specify the maximum number of transactions, storage requirements, and any other quantifiable information about capacity requirements placed on the system. Estimate when these capacities are expected to draw on department resources. Make distinctions for changes to capacity limits because of varying modes of operation.

Determine Accuracy and Validity Requirements

Describe accuracy requirements to be imposed on the system, such as mathematical calculations. Impose validity requirements on the system, such as the following:

- Check fields to ensure correct data type (e.g., numeric, alpha, alpha-numeric, date, time, and logical).
- Check for presence of required data.
- Validate database element codes.
- Identify and describe interfield relationships.

Identify Input and Output Effects on Performance

Describe the data input and output and their effects on performance. Include the following data types:

- Reports and queries to be generated by the system
- Online input, including data from presently used manual forms
- Interfaces from other systems

Define Failure Contingencies

Discuss alternative courses of action that may be taken to satisfy information requirements if the proposed system fails. Include, as appropriate, backup, fallback, and degraded modes of operation.

In the event that a primary system element is dysfunctional, describe backup requirements for ensuring continued achievement of system functions. Include backups for restoration of data so that restarts can be done. An example of a backup procedure would be checkpoint restart capabilities in batch programs with runtimes exceeding one hour.

Fallback indicates the use of another system or means to satisfy system objectives. Explain fallback techniques for ensuring the continued satisfaction of specific system objectives. Also discuss the potential impact of a system fallback on interfacing systems. An example of fallback for an automated system might be manual manipulation and recording of data.

For degraded modes of operation, state priorities for restoring the essential processing steps in the event that full processing capability is not available.

2.2.3 Define Operational Environment

Define the operational environment in which the system will be required to perform. Include such characteristics as system application interfaces, software and hardware constraints, estimated input and output peak volumes, proposed hardware and software configuration, and resolution of hardware and software failures.

Identify System Application Interfaces

Describe all interfaces with other application systems and subsystems. For each interface, the following information should be specified:

- Description of operational considerations of data transfer, such as security
- General description of data transfer requirements to and from the system
- Characteristics of communications media or systems used for transfer

- Format, unit of measurement, range of values, and data codes
- Type of anticipated interface
- Anticipated interface procedures

Define Software and Hardware Constraints

Identify any constraints that will affect operation of the system. Define any limitations affecting desired capability (e.g., availability of main memory and storage devices). Include any desired capabilities that will not be provided by the proposed system and the reason(s) why each will not be provided. Also include any anticipated operational changes that will affect proposed operation of the system.

Estimate Input and Output Peak Volumes

Determine maximum number of transactions that will be input to the system during a specified time frame and maximum number of outputs expected from the system during a specified time frame.

Estimate Required Hardware and Software Configuration

Determine hardware and software expected to be required to operate the proposed system. If known at this point, include such software as languages, compilers, the operating system, report and code generators, database management systems, and text editors. When determining what programming language will be used, consider the following:

- Knowledge of language within HUD and among its contractors
- Vendor support
- Ease of maintenance
- Machine resources available
- Availability of support and utility software for the language(s)
- Capability to run on proposed system platform(s)
- Support for strong system engineering principles
- Compatibility with DBMS
- Generation by the CASE tool being used by the project
- Compliance with HUD standards

Include such hardware items as number and type of storage devices required, specialized input devices, and number and type of printers required. If these hardware and software items are not currently in place, document the plan for acquiring them in the System Support Plan.

Address Hardware and Software Failures

Discuss possible failures of the hardware or software systems, the consequences (in terms of system performance) of such failures, and the alternative courses of action that may be taken to satisfy information requirements.

2.2.4 Identify Control Objectives

Describe the control objectives the system is intended to support, including input, output, and process control points; vulnerabilities associated with control points; administrative and physical safeguards of each control point; and audit trail requirements.

Identify Input, Output, and Process Control Points

Define input, output, and process control points. A control point can be located at any interface where movement of data occurs within or between sites.

For input control points, identify the following:

- *Origin*: The point at which input data will be collected and prepared for entry into the system.
- *Data entry*: Any remote devices intended to be used to perform data entry, updates, and corrective actions.
- *Disposition*: The disposition of source data after it is entered into the system.
- *Error correction*: Points at which data input errors will be detected, reported, and corrected.

For process control points, identify the following:

- *Accuracy and completeness*: Points at which the system will provide notification of success or failure of the requested processing.
- *System interfaces*: Points at which the system will pass data to or receive data from other systems.

For output control points, identify the following:

- *Production*: Types and locations of devices authorized to receive output.
- *Distribution*: Steps involved in distributing and deploying output products.

Describe Vulnerabilities by Control Point

Define the vulnerabilities by control point. A system vulnerability is a design, implementation, or operational condition inherent in the application or system that lends itself to error, data loss, compromise of information, unauthorized dissemination of data, or denial of service.

Describe Administrative and Physical Safeguards by Control Point

Describe the safeguard requirements (administrative and physical) at each control point to reduce the vulnerabilities. An administrative safeguard is defined as any procedure that requires management supervision and includes any of the following:

- Personnel positions requiring security clearances or access authorization
- Any requirements for proper control of collection, preparation, and backup of data
- Requirements to limit operation of the system to certain devices or periods of the day
- Requirements for a variance from standard distribution procedures

- Procedural requirements to develop, maintain, and control access or permissions to system data or functions (i.e., different levels of access or process authority)

A physical safeguard is any physical means that limits access to data, such as locked doors, vaults, and card-key access. Describe requirements for dedicated equipment as well as on-site and off-site storage.

Describe Audit Trail Requirements

Describe all user requirements for an audit trail, such as total transactions processed by location, time, and type. At a minimum, the requirements should include those specified in the HUD ADP security plan and standards (Handbook 2400.24) and guidelines.

2.2.5 Identify Cost Constraints

Identify cost factors that may limit design, development, and continued operation of the proposed system. Include personnel and resource cost estimates, as well as costs associated with equipment and system operations.

2.2.6 Document Functional Requirements

Accumulate information developed during the effort to define system functions. Document all information regarding functional capabilities of the system in accordance with HUD SDM documentation standards (Handbook 2400.15). The document will take the form of a Functional Requirements Document.

Document the derived functional requirements in a requirements matrix to show the relationship of approved system objectives to the functional requirements. Use a tool, such as a spreadsheet or word processor table, to manage the matrix. Insert the matrix into the Functional Requirements Document.

2.3 DETERMINE DATA REQUIREMENTS

Data requirements provide a detailed description of the data model that the system must use to fulfill its functional requirements. Users and developers work jointly to identify requirements and with HUD Data Administration for defining the domain data model. In situations where users and Data Administration determine the model independently of developers, hold walkthroughs during the identification so that users can describe the requirements and the model to developers and receive feedback about the clarity and completeness of requirements. Separate the data description into two categories: static and dynamic data. Arrange data elements in each category in logical groupings, such as functions, subjects, or other groupings most relevant to their use. Describe the type of information required to document the characteristics of each data element. Specify information, including that related to sensitivity and privacy issues, to be collected by the user and developer. Include the following activities:

2.3.1 Categorize and define the data.

2.3.2 Define data constraints.

2.3.3 Identify input responsibilities.

2.3.4 Identify data collection requirements.

2.3.5 Document data requirements.

Table 2-5 defines roles and responsibilities of key personnel for determining the data requirements.

Table 2-5. Roles and Responsibilities for Determining Data Requirements

Role	Responsibility
Project Sponsor	Provides input and approves data requirements.
User	Works jointly with developer to determine data requirements and dependencies.
Developer	Works jointly with the user to determine data requirements and dependencies. Prepares Data Requirements Document.
Computer Services	Participates in review of Data Requirements Document, and uses information for capacity planning.
ADP Security	Provides input to the data requirements analysis (optional).
Project Data Administrator	Supports generation of application requirements. Adds application-specific entities or objects to the application data model originally taken from the enterprise-wide data model. Generates and revises the problem domain data model. Reviews and reconciles models; ensures that data elements conform to HUD naming standards. Participates in review of Data Requirements Document.
Quality Assurance	Monitors and reviews data requirements activities and products.

2.3.1 Categorize and Define the Data

For both static and dynamic data requirements, organize defined data elements into logical records and describe the relationships among these logical records. Organize the data in a way that facilitates processing of inputs to generate outputs and satisfies other computational requirements.

Identify Static and Dynamic Data Requirements

Static data are defined as data primarily used for reference during operation and usually generated or updated in widely separated time frames, independent of normal software execution. Dynamic data include all data intended to be updated and either input during normal execution or output.

List static data elements used for either control or reference purposes. List dynamic input data elements that constitute data intended to be changed during normal system execution or online operation. For both categories of data, work with the Project Data Administrator to model data elements in logical groups, such as functions, subjects, or other groups that are most relevant to their use. Include the following for each data element:

- Data element name
- Synonymous name
- Type (e.g., alpha, alpha-numeric, or numeric)
- Definition
- Format
- Range of values or discrete values

Return to Title Page

- Unit of measurement
- Precision (e.g., number of decimal places)
- Data item names, abbreviations, and codes
- Characteristics, such as precision, accuracy, timing, and capacity

Identify Internally Generated Data

Identify data created as a result of program calculations or other program manipulations, such as algorithms.

2.3.2 Define Data Constraints

Identify any restrictions on each data element that must be considered during definition of the system. Limitations may be placed on the data because of such factors as the source of data input, devices used for input and output of data, recipients of the data, the converted form the input and output data will take, and how often the data will be updated.

Identify Source of Input

Identify the source from which each data element will be entered, such as an operator, workstation, or organizational unit, including form or medium. Describe all sources from which data will be gathered, including organizations internal and external to HUD, automated systems that will directly interface with the application under development, and those systems that will transmit data to the new system.

Identify Input and Output Medium and Device

Identify the medium and hardware intended for entering each data element into the system. In situations in which only specific workstations are to be legitimate entry points, so specify.

Identify the medium and hardware device intended for presenting output data to the recipient. Specify whether the recipient is to receive the data as hardcopy, screen display, or another medium. If the output is to be passed to another automated system, specify the medium.

Identify Data Recipients

Name the organization that will be receiving each output data element (in any form, such as data file or report).

Specify Conversion Factors

If any stored data are measured in one unit of measure but processed by the system in another unit of measure, explain the data conversion process. For example, a data element is a function of time, stored in hours; the new system requires that the data be stored in terms of minutes. In this example, the conversion factor is “multiply by 60.”

Specify Frequency of Update and Processing

State the expected frequency of data element change and the expected frequency of processing input data elements. If the input arrives in a random manner, specify both the average frequency and some measure of the variance.

2.3.3 Identify Input Responsibilities

Determine who will be responsible for input of data. Consider establishing a data input group, if necessary. This group would collect the data, put them in the correct format, and perform the data entry function.

2.3.4 Identify Data Collection Requirements

Describe procedures that will be used to collect data, including a detailed format for the input data, how the data will be transmitted to the system, and how frequently the data will be input.

Describe Detailed Formats

Describe, in detail, the format of data to be input to and output from the proposed system. Descriptions may be narrative or in the form of data record layouts and screen formats, as appropriate.

Identify Data Communication Media

Determine the devices that will be used to transmit data to and from the proposed system. Identify each device by name and provide a brief description of the process used to transmit the data.

Determine Timing of Input

Identify the frequency with which data are input to the proposed system for each input data requirement. This frequency may be expressed using terms such as daily, weekly, monthly, yearly, each minute, hourly, or continuously.

2.3.5 Document Data Requirements

Document the information gathered regarding the system's data requirements in accordance with HUD SDM documentation standards (Handbook 2400.15). The document will take the form of a Data Requirements Document.

Document the derived data requirements in a requirements matrix to show the breakdown of system objectives to data. Use a tool to manage the matrix. Insert the matrix into the Data Requirements Document.

2.4 DEVELOP SYSTEM SECURITY PLAN

The objective of the System Security Plan is to improve protection of information and information processing resources for HUD's sensitive application systems. Its purpose is to provide a basic overview of the security and privacy requirements of the project and HUD's plan for meeting those requirements. The System Security Plan also may be viewed as documentation of the structured process of planning adequate, cost-effective security protection for a system. Therefore, it should reflect input from all organizational areas with responsibilities concerning the system. Activities include the following:

2.4.1 Prepare system identification information.

2.4.2 Determine sensitivity of information.

2.4.3 Identify system security measures.

2.4.4 Prepare additional comments.

2.4.5 Document security information.

Table 2-6 defines roles and responsibilities of key personnel for developing the System Security Plan.

Table 2-6. Roles and Responsibilities for Developing the System Security Plan

Role	Responsibility
User	Assists in defining access and data rights.
Developer	Develops System Security Plan.
ADP Security	Assists in development of the System Security Plan (optional).

2.4.1 Prepare System Identification Information

Identify the following information about the proposed system:

- Responsible organization
- System name or title
- System code
- System category
  - *Major application*: performs clearly defined functions for which there is a readily identifiable security consideration and need
  - *General support system*: provides general ADP or network support for a variety of users and applications
- Operational status
  - *Operational*
  - *Under development*
  - *Undergoing a major modification*
- General description
- System environment and special considerations
- Information point(s) of contact

2.4.2 Determine Sensitivity of Information

Include the following information for this activity:

- Applicable laws or regulations affecting the system
- Description of information sensitivity, indicating which of the following protection requirements apply:
  - *Confidentiality*: system contains information that requires protection from unauthorized disclosure

  - *Integrity*: system contains information that must be protected from unauthorized, unanticipated, or unintentional modification
  - *Availability*: system contains information or provides services that must be available on a timely basis to meet mission requirements or to avoid substantial losses
- Description of the information handled by the system and the need for protective measures
- Estimate of the risk resulting from loss, misuse, or unauthorized access to or modification of information in the system
- For each of the protection requirements, indicate one of the following risk levels:
  - High (critical concern of the system)
  - Medium (an important concern but not necessarily paramount in the organization's priorities)
  - Low (requires some minimal level of security but not to the same degree as the previous two categories)

2.4.3 Identify System Security Measures

Describe control measures, in place or planned, intended to meet protection requirements of the system. Types of control measures should be consistent with the need for protection of the system, as described earlier.

Include the following information for this activity:

- Protection requirements to control the risks
- Specific standards used in design, implementation, or operation of protective measures
- Security control descriptions indicating whether each is in place, planned, or not applicable

For major applications, also include the following:

- Overall management controls
- Procedures to assure protection is built into the system
- Day-to-day procedures and mechanisms to protect systems when operational
- Security awareness and training
- Hardware and software controls used to provide automated protection or to facilitate manual protection
- Complementary controls provided by support systems

For general support systems, include the following:

- Management controls
- Acquisition, development, and installation controls

- Day-to-day procedures and mechanisms to protect operational controls
- Security awareness and training
- Controls to protect the system from unauthorized access or misuse
- Controls over the security of applications

2.4.4 Prepare Additional Comments

Include additional comments about the security of the subject system and any perceived need for guidance or standards.

2.4.5 Document Security Information

Accumulate system security information and prepare a System Security Plan document in accordance with HUD SDM documentation standards (Handbook 2400.15).

2.5 DEVELOP SYSTEM AUDIT STRATEGY

The Inspector General's staff usually develops the Internal Audit Plan, encompassing all agency system activities. Systems under development may be selected for review based on several factors, including the sensitivity or criticality of the system or the effectiveness of internal agency information system management control. For those systems selected for audit review, detailed specific plans are prepared. The plan clarifies audit involvement, which may range from review of completed work products to active audit participation in each system development activity. In any case, the overall objective is to assess the adequacy of internal data processing control and provide reasonable assurance to management that adequate controls are in place.

The project manager formally notifies the IG staff of the existence of the information system project and provides all project information requested by the IG staff. Requested information could include the Project Plan and the System Security Plan. The IG staff prepares the Internal Audit Plan; the original plan is maintained by the IG and a copy is included in the central project file.

Table 2-7 defines roles and responsibilities of key personnel for developing the System Audit strategy.

Table 2-7. Roles and Responsibilities for Developing the System Audit Strategy

Role	Responsibility
Project Manager	Coordinates with IG staff.
IG Staff	Coordinates with project manager, as necessary. Develops the Internal Audit Plan to incorporate information received from the project manager.

2.6 UPDATE SYSTEM DECISION PAPER

Update the System Decision Paper to include a summary of the progress made during the Define System phase and the schedule of events for Define System phase activities. Include the

Return to Title Page

current status of the project. List in detail any changes to the original plan for the development of the project and the parties responsible for approving those changes. The updated System Decision Paper is then reviewed and approved by the appropriate management officials. Include the following activities:

2.6.1 Summarize progress of system.

2.6.2 Identify changes.

2.6.3 Summarize schedule of events and status.

2.6.4 Document results.

Table 2-8 defines roles and responsibilities of key personnel for updating the System Decision Paper.

Table 2-8. Roles and Responsibilities for Updating the System Decision Paper

Role	Responsibility
Project Manager	Updates System Decision Paper as necessary.
User	Provides inputs needed to update System Decision Paper.
Developer	Provides inputs needed to update System Decision Paper.
Quality Assurance	Reviews System Decision Paper for adherence to standards.

2.6.1 Summarize Progress of System

Summarize the progress of the project through the Define System phase, and update the System Decision Paper to include this summary.

Identify Development Alternative Selected

Briefly describe the selected development alternative, and provide a justification for the selection of this alternative. Update the System Decision Paper to include this information.

Identify Resources Expended, Needed, and Projected

Identify resources that were expended during the Define System phase, and update the projected schedule of resources to be expended through the remaining system life cycle.

2.6.2 Identify Changes

Document any changes to the system development milestones and schedule or any modifications made to the project strategy. Update the System Decision Paper accordingly.

Describe Changes in Milestones, Schedules, and Activities

Provide a description of any Define System phase occurrences that may have had an impact on the projected milestones, schedules, and activities of the development effort. Provide a brief explanation of the nature of these impacts and describe any changes to the milestones or schedule that may occur as a result of these changes.

Return to Title Page

Describe Modifications to Project Strategy

Provide a description of any modifications to the project strategy that have occurred during the Define System phase of development. Provide a brief explanation of these modifications, indicating the reasons for the changes as well as any anticipated impacts.

2.6.3 Summarize Schedule of Events and Status

Develop a brief summary of the events that took place during this phase of development and a description of the remaining schedule of development activities. Show the impact these events have had on the original schedule and any anticipated impacts on future development.

2.6.4 Document Results

Document all information for this phase in the System Decision Paper in accordance with HUD SDM documentation standards (Handbook 2400.15).

2.7 REFINE PROJECT PLAN

The Project Manager controls the project by monitoring phase activities, taking corrective action where necessary, and refining the project plan to account for changes attributable to actions taken in the current phase and new information for upcoming phases. The Project Manager also reviews the quality process planned by users, developers, testers, and quality assurance for the next phase. Activities include the following:

2.7.1 Update Project Plan.

2.7.2 Review planned quality process.

Table 2-9 defines roles and responsibilities of key personnel for refining the Project Plan.

Return to Title Page

Table 2-9. Roles and Responsibilities for Refining the Project Plan

Role	Responsibility
Project Sponsor	Approves changes to project schedule. Approves changes to Project Plan.
Project Manager	Updates Project Plan to include actual costs incurred and specific activities accomplished for the Define System phase; revises project schedule, plans, strategies, and resources and requirements for subsequent phases, as required. Reviews, for approval, the quality process planned by users and developers for the next phase.
User	Provides updated input to Project Manager on required Design System phase activities. Works with developer to review and update the quality process planned for the next phase. Attends review of Project Plan to provide technical input to the review process, as required.
Developer	Provides updated input to Project Manager on required Design System phase activities. Works with user to review and update the quality process planned for the next phase. Participates in review of Project Plan.
Computer Services	Participates in review of Project Plan.
Quality Assurance	Provides updated input to Project Plan.
Configuration Management	Provides updated input to Project Plan.

2.7.1 Update Project Plan

Update the Project Plan with cost, schedule, and budget data for the current phase (Define System) to the level of detail necessary to reflect the status of the project. Review the plan and determine if any activities described for the next phase (Design System) and subsequent ones are affected by the completion status of the current phase activities. Adjust the schedules and resource requirements for activities in the next phase, if necessary, and assign starting and ending dates for activities that have been affected. Take into account that the starting date for some activities may be dependent on the completion of other activities. Update milestones, schedules, and resource requirements for the remainder of the project.

2.7.2 Review Planned Quality Process

Users, developers, testers, and QA work closely together to determine the process to be used to build the product with the desired quality. Activities include tailoring the planned quality process for the next phase and identifying the standards and procedures to be used. The project manager reviews the quality process for approval and execution in the next phase.

2.8 REVIEW AND APPROVE DOCUMENTS AND DELIVERABLES

Review and approval of documents and deliverables is an iterative process as each phase provides more definition and as products are revised. In addition, the approval process offers senior management the capability to monitor the activities of the project. Activities include the following:

2.8.1 Review revised documents from prior phases, as required.

2.8.2 Conduct reviews for Define System phase documents and deliverables.

2.8.3 Obtain approval of project documents and deliverables.

Table 2-10 defines roles and responsibilities of key personnel for reviewing and approving documents and deliverables.

Table 2-10. Roles and Responsibilities for Review and Approval of Deliverables and Documents

Role	Responsibility
Project Sponsor	Participates in reviews (optional).
Project Manager	Determines needed changes to prior phase documents and ensures revision in accordance with the overall Project Plan. Coordinates review of Define System phase deliverables. Determines level of required review and schedules required reviews. Attends reviews and presents deliverables. Ensures that Define System phase deliverables are updated to include any recommendations received during the review. Follows up on reviewers' recommendations for revised documents. Obtains appropriate concurrence and approval for Define System phase document deliverables and associated Management Summaries. Obtains appropriate concurrence and approval of revised documentation.
User	Revises prior phase products when applicable. Assists the project manager with preparation of Define System phase deliverables for review. Attends deliverable reviews.
Developer	Revises prior phase products when applicable. Assists the project manager with preparation of Define System phase deliverables for review. Provides technical expertise during the review process.
ADP Security	Reviews Define System phase deliverables to ensure that all necessary security requirements have been incorporated. Reviews revised products to ensure that the necessary security standards and guidelines have been addressed.
Quality Assurance	Reviews revised products to ensure they meet all applicable HUD standards and guidelines. Ensures that all review procedures are followed, as required. Reviews Define System phase deliverables to ensure they meet all applicable HUD standards and guidelines.
Configuration Management	Assists project manager with review preparation.

2.8.1 Review Revised Documents from Prior Phases, as Required

Review the activities performed during the Define System phase to determine if they have an impact on any documents produced during the previous phases. If changes are required to prior lifecycle phase documents, update those documents to reflect current project developments. The revised documents are reviewed by project personnel similarly to reviews for current phase documents to ensure that changes are within the scope of the project's requirements and are in compliance with HUD and project standards and procedures. Prepare a management summary for each revised document that summarizes the essential revisions. Submit the revised products to the appropriate review board for approval. File affected product change records with CM for appropriate version control updates. Use the appropriate document review checklist in Appendix E to aid in the document reviews.

2.8.2 Conduct Review for Define System Phase Documents and Deliverables

Conduct a project review with project personnel and system stakeholders to ensure that the project documents and deliverables for the Define System phase include the necessary level of detail, fulfill the system's requirements, and meet the appropriate HUD and project standards and guidelines. Use the appropriate document review checklist in Appendix E for Define System phase deliverables to aid the review process.

A minimum of 10 working days before the scheduled review, notify the personnel required to attend and provide each with a copy of the product for their pre-review. Discuss all comments or objections that are raised during the review, and reach a consensus on one of the following before the review session terminates:

- Document is correct and complete, as is, without any further changes.
- Additional changes need to be made, but are minor and do not require further review. In this case, the updates should be made and change pages should be distributed by an agreed-upon date.
- Required changes will have a major impact on the plan. A second review must be scheduled. The changes must be incorporated and the resulting change pages distributed at least 10 working days before the second review.

Prepare a management summary for each document that summarizes the essential data collected in the document, conclusions that may be drawn from the document, and their potential impact on the project, if applicable. Submit the management summaries to the appropriate review board for approval. Documents for approval include the following:

- Functional Requirements Document
- Data Requirements Document
- System Security Plan
- System Decision Paper (updated)
- Project Plan (updated)

2.8.3 Obtain Approval of Project Documents and Deliverables

Present the project documents and deliverables to the chairperson of the appropriate review board for approval at least 10 days before the scheduled decision date. Include the management summary information, approval (sign-off) record, the System Decision Paper, and any other required or requested information. The review board chairperson coordinates review board comments, recommendations, and approval signature(s), and returns the approval record to the project manager. Recommendations on the approval record are addressed by the project and the document should be resubmitted to the board if requested. Approval will be assumed if there is no response or if the response is “no comment.” Project approval records are maintained by the project’s configuration management (CM) function. A copy is inserted into the central project library.

The project proceeds to the next phase after all project documents and deliverables are approved.

2.9 PERFORM CHANGE CONTROL ACTIVITIES

Any products developed during Define System phase are baselined and subjected to version control. Products baselined during prior activities are assigned new version control numbers when they undergo change (e.g., an update or rewrite). Configuration management reports are provided to the project manager as requested or required.

Table 2-11 defines roles and responsibilities of key personnel for performing CM.

Table 2-11. Roles and Responsibilities for Performing Change Control Activities

Role	Responsibility
Project Manager	Approves controlled products for distribution.
Configuration Management	Performs change control for new and revised products. Provides status reports to project management as requested or required.
Quality Assurance	Audits products to ensure that only approved changes are addressed.

The CM function manages, implements, and reports on project CM activities. In supporting the project, CM performs the following activities:

- ***Verify changes made to product.*** Review the updated product to ensure that the changes have been made as described in supporting documentation. Supporting documentation may be comments received from document reviews. Supporting documentation for software may be software change requests or discrepancy reports generated during testing.
- ***Assign version number.*** The version number must follow conventions established by HUD and enable the project's CM to monitor updates to the product and assist in its distribution. For software programs, increment the version number each time a change is made to the software to correct deficiencies found during testing.
- ***Store approved version in central library.*** After each baseline product has been completed and approved, according to HUD procedures, store the approved version in the project’s central library. The project's CM function controls access to the library. For software, the library can be a subdirectory or dataset where all baselined software configuration items (CIs) will be stored.
- ***Record product information in inventory log.*** Maintain an inventory log that includes the title of the product, release date, version number, name and version or model numbers of the software and hardware used in the development of the

Return to Title Page

product, name of the organization responsible for development of the product (usually the sponsoring organization), and the product distribution list.

- ***Distribute copies of products as required.*** The project's CM function distributes copies of the products according to a distribution list maintained as part of the inventory log information, with distribution based on need and the security level of the product.
- ***Archive old versions of products.*** Archive and retain outdated versions of all products for the required period of time, in keeping with HUD standards and guidelines.

This page intentionally left blank.

Return to Title Page

Section 3. Design System

Return to Title Page

3.0 Design System

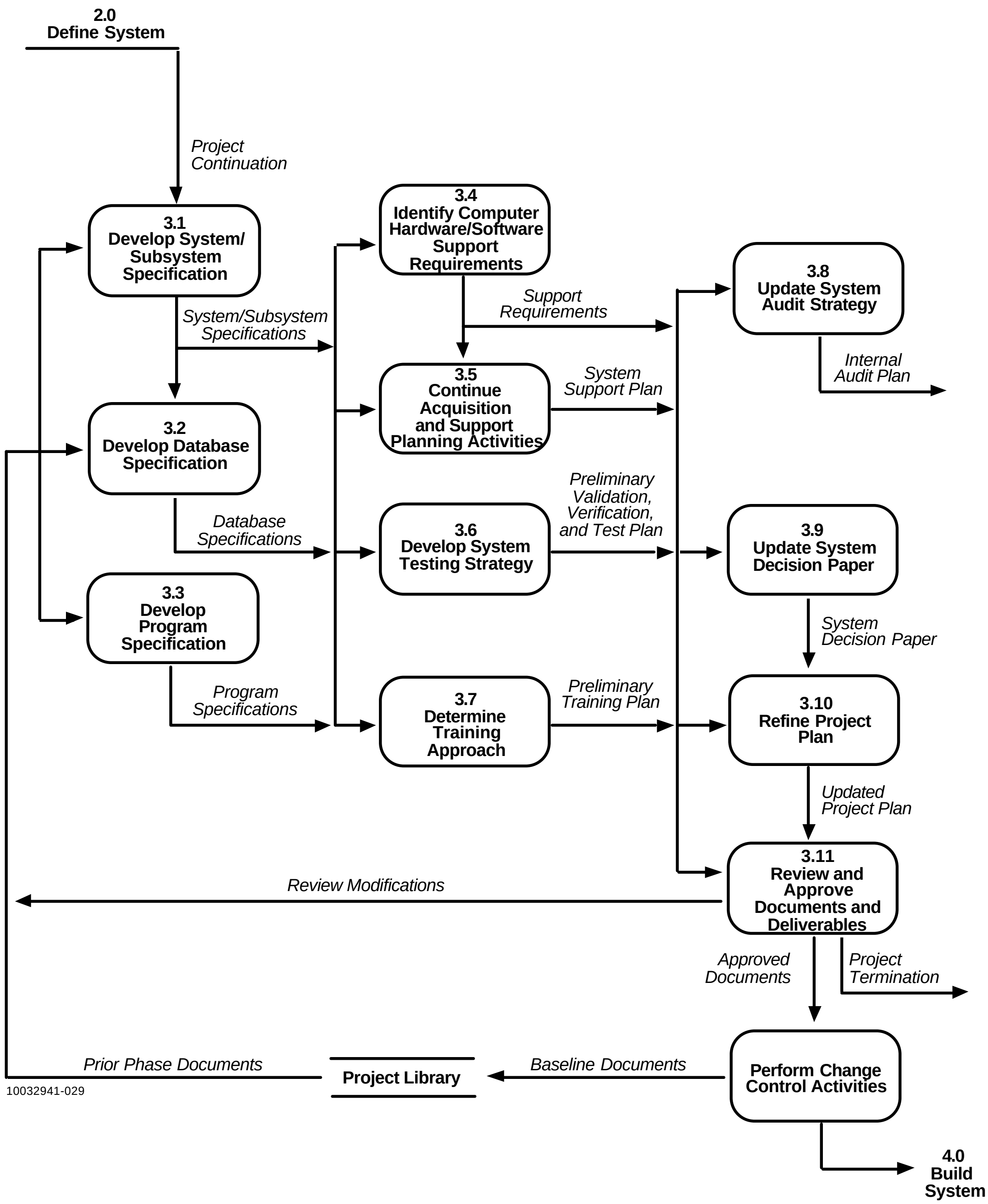

Figure 3-1. Process Flow for Design System Phase

3.0 DESIGN SYSTEM

Purpose

The purpose of the Design System phase is to develop detailed specifications that emphasize the physical solution to the user's information management need as described in the Define System documentation. The detailed specifications are used in the Build System phase to create the system. Figure 3-1 highlights the process flow for the Design System phase.

Overview

After the functional and data requirements are documented in the Define System phase, they are further refined to low-level specifications (system/subsystem, program, and database specifications). The specifications are then organized in a way suitable for implementation within the constraints of a physical environment (e.g., computer, database, facilities). The Functional Requirements Document and the Data Requirements Document are used to develop the System/Subsystem Specifications, Program Specifications, and Database Specifications documents.

The project development team uses the functional and data requirements from the Define System phase for partitioning the system into subsystems. Based on the physical constraints, the requirements are refined into low-level functions and system/subsystem specifications and documented in the System/Subsystem Specifications document. In refining the requirements, the development team allocates the system processes and associated required data to subsystems based on the following considerations:

- System/subsystem functions
- Accuracy and validity imposed on the system/subsystem
- Timing requirements placed on the system/subsystem
- Adaptability of system/subsystem to changing requirements
- Equipment needs
- Communications environment
- Support software
- System security and controls
- Input and output records

The project development team uses the lowest level functions identified in the System/Subsystem Specifications document to develop programs. Based on the physical constraints, the system/subsystem specifications are refined into program specifications and are documented in the Program Specifications document. In refining program specifications, the development team allocates the program processes and associated required data to programs based on the following considerations:

- Functions of the software units
- Accuracy and validity imposed on the software units
- Timing requirements

- Support software with which the software units must interact
- Interfaces with other application software
- Storage requirements
- Security requirements
- Input and output data and output reports
- Data retention
- Logic of each software unit

The project development team uses the data requirements and data models from the Define System phase to develop the databases. Based on physical constraints, the requirements and models are refined into database specifications and are documented in the Database Specifications document. In refining database specifications, the development team allocates data requirements and models to databases based on the following considerations:

- Unique database names
- Special instructions for database usage
- Support software requirements
- Security considerations, sensitivities, and critical issues
- Database management system configuration
- Database schema

The development and system acceptance test teams develop the initial strategy for testing the system. This strategy includes plans for unit and integration tests as well as the system acceptance tests that will be performed during the Evaluate System phase. The project manager estimates the required resources to support the test strategy and associated test plans.

The training needs of the system acceptance test team and project development team as well as those of the operations personnel are considered, and plans are developed to produce the necessary training modules and manuals to meet these needs. Any impact these plans may have on the system development schedule is recorded in the Project Plan.

The project manager updates the estimates of project expenses for this phase to reflect actual costs and the estimates for the future phases. In addition, the work planned for future development phases is redefined based on information acquired during this phase. The project manager updates the Project Plan to reflect any changes in the development effort tasks and activities that occur as a result of Design Phase efforts.

The development team and project manager review all deliverables, as depicted in Table 3-1. They review the Design System phase functions and products before the end of the phase or phase review.

The project manager presents the System Decision Paper and all required supporting documentation to the appropriate review board at the end of this phase of development. Projects that receive the review board’s approval continue to the next phase of the development lifecycle, the Build System phase.

Return to Title Page

Table 3-1. Design System Phase Functions and Products

Design System Functions	Products
3.1 Develop System/Subsystem Specifications	System/Subsystem Specifications
3.2 Develop Database Specifications	Database Specifications
3.3 Develop Program Specifications	Program Specifications
3.4 Identify Computer Hardware and Software Support Requirements	--
3.5 Continue Acquisition and Support Planning Activities	System Support Plan (updated)
3.6 Develop System Testing Strategy	Verification, Validation, and Test (VV&T) Plan
3.7 Update Audit Strategy	Internal Audit Plan (updated)
3.8 Determine Training Approach	Training Plan
3.9 Update System Decision Paper	System Decision Paper (updated)
3.10 Refine Project Plan	Project Plan (updated)
3.11 Review and Approve Documents and Deliverables	New and revised products Management Summary[1]
3.12 Perform Change Control Activities	Change control records

[1] A management summary is prepared for each product produced or revised during the Design System phase. This one-page summary includes a summary of the essential data collected in a document product, conclusions that may be drawn from the document, and potential impacts on the project, if applicable.

Standards and Guidelines

The HUD SDM documentation standards (Handbook 2400.15) and project guidelines should be followed during development of Design System phase products.

Roles and Responsibilities

Throughout the Design System phase of development, key personnel are required to perform various activities. Table 3-2 lists types of personnel required and the activities for which they are responsible.

Table 3-2. Roles and Responsibilities for Design System Phase (1 of 3)

Role	Responsibility
Project Sponsor	Approves project schedule revision. Approves Project Plan modifications. Designates independent system acceptance test team. Participates in Design System phase reviews (optional).
Project Manager	Continues acquisition support activities. Coordinates system support activities.

Return to Title Page

Table 3-2. Roles and Responsibilities for Design System Phase (2 of 3)

Role	Responsibility
Project Manager (Cont'd)	Oversees formation of system acceptance test team. Coordinates review of Design System deliverables. Updates Project Plan to include actual costs incurred and specific activities accomplished for the Define System phase; revises project schedule, plans, strategies, resources, and requirements for subsequent phases, as required. Reviews, for approval, the quality process planned by users and developers for the next phase. Updates Design System deliverables to include any recommendations received during review. Obtains appropriate concurrences and approvals for Design System document deliverables and associated management summaries. Coordinates with IG staff, as necessary. Determines needed changes to prior documents, and ensures revision in accordance with overall Project Plan. Obtains appropriate concurrences and approvals of updated documentation. Determines schedule and level of required reviews. Attends reviews and presents deliverables. Updates System Decision Paper, as necessary. Coordinates capacity planning with Computer Services.
Project Development Team (User and Developer)	Determines additional resource and support requirements. Prepares System/Subsystem Specifications. Prepares Database Specifications and Program Specifications. Identifies hardware and software support requirements. Updates requirements matrix. Provides input for development test strategy for VV&T Plan. Provides input to project manager on required Design System tasks and activities. Assists in revision of prior products. Assists project manager with preparation of Design System deliverables for review. Attends Design System phase deliverable reviews. Provides technical expertise during review process. Reviews and updates the quality process planned for the next phase.
System Acceptance Test Team	Develops system testing strategy. Prepares preliminary VV&T Plan. Identifies test resources and test evaluation criteria. Describes testing methods and test tools.

Return to Title Page

Table 3-2. Roles and Responsibilities for Design System Phase (3 of 3)

Role	Responsibility
Computer Services	Confirms/approves additional project support requirements. Provides updates to System Support Plan. Reviews input to system testing strategy. Provides input to training approach. Assists in determining production platform. Participates in review of Design System phase documents and deliverables. Reviews capacity planning estimates with project manager.
ADP Security	Reviews Design System deliverables to ensure that all necessary security requirements have been incorporated. Reviews revised products to ensure that necessary security standards and guidelines have been addressed.
IG Staff	Coordinates with project manager, as necessary. Updates internal audit plan to incorporate information received from project manager.
Project Data Administrator	Defines database design specification. Assesses design of logical data model. Participates in and assesses design of physical data model. Supports preparation of database specification. Reviews Design System phase documents and deliverables.
Project Database Administrator	Participates in review of Database Specifications.
Configuration Management	Establishes change control over new products. Performs version control for revised products. Provides configuration management reports to project management, as requested or required. Assists the project manager with preparation of Design System deliverables for review.
Quality Assurance	Reviews revised products to ensure they meet all applicable HUD standards and project guidelines. Ensures that all review procedures are followed, as required. Reviews Design System deliverables to ensure they meet all applicable HUD standards and project guidelines. Coaches users and developers in the planning for quality in the next phase.

3.1 DEVELOP SYSTEM/SUBSYSTEM SPECIFICATIONS

The primary focus of this activity is to develop and describe the detailed design of the system/subsystems. The development team analyzes the functions, described during the Define System phase, and determines which functions logically may be processed together to form a subsystem. The development team documents the system/subsystem specifications that are used to partition the system/subsystem and hierarchically decompose it to the lowest-level functions. The system/subsystem specifications comprise the system detailed design. The following activities are performed to develop the system/subsystem specifications:

Return to Title Page

3.1.1 Analyze functional requirements and data requirements.

3.1.2 Describe system/subsystem specifications.

3.1.3 Perform walkthroughs

3.1.4 Document system/subsystem specifications

Table 3-3 defines the roles and responsibilities of key personnel for acquiring and installing system components.

Table 3-3. Roles and Responsibilities for Developing System/Subsystem Specifications

Role	Responsibility
Project Manager	Coordinates development of system/subsystem specifications.
User	Defines system/subsystem specifications.
Developer	Defines and prepares system/subsystem specifications.
Project Data Administrator	Analyzes data requirements.
Quality Assurance	Reviews system/subsystem specifications for adherence to standards.

3.1.1 Analyze Functional Requirements and Data Requirements

The project development team performs a detailed study of the functional and data requirements developed during the Define System phase to determine which requirements and data logically may be processed together. The logical groupings of requirements and data are subsystems.

3.1.2 Describe System/Subsystem Specifications

Develop a description of the system/subsystem specifications. If a prototype was developed earlier in the lifecycle, include those specifications defined with the prototype. For each system/subsystem, include such information as timing requirements, equipment needs, communication environment, support software, security, and input and output records. During the process of describing the system/subsystem specifications, the project development team performs the following actions:

- Partition the functional and data requirements into logical subsystems.
- Document the traceability between the functional and data requirements and the defined subsystems.
- Characterize the features incorporated in the design of the system/subsystem that meet the accuracy and validity requirements imposed on the system/subsystem during the Define System phase.
- Determine the timing requirements imposed on the system/subsystem under varying conditions.
- Determine the system/subsystem's capabilities for adapting to changing requirements, and document how these capabilities are incorporated into the system/subsystem's

design. Identify those procedures and components designed to support failure contingencies.

- Identify the equipment needed for the operation of the system/subsystem. Include a description of the equipment that is currently available as well as the characteristics of any new equipment requirements that may be necessary for the system/subsystem to operate correctly.
- Identify the system/subsystem's data communications environment. Develop detailed schematics of the portions of the communications environment that directly relate to the system/subsystem.
- Identify the software with which the system/subsystem must interact. Include both support software and test software, if needed. Provide the correct name, description, and documentation references of each such software system, subsystem, and program. Include a reference to the language (compiler, assembler, program, query) and to the operating system to be used. If operation of the computer programs to be developed is dependent on forthcoming changes to other software, identify and discuss the nature, status, and anticipated availability date of such changes.
- Outline the potential threats to the system/subsystem and the security measures that have been incorporated to reduce or eliminate those threats. An example would be the threat of unauthorized access being reduced by incorporating into the design a password requirement at all system/subsystem access points. List and describe all controls that have been incorporated in the system/subsystem design; for example, batch controls on the entry of financial data.
- Identify the input and output data and reports. For inputs, provide information about the characteristics of each input to the system/subsystem, such as file name, format and type of data, validation criteria, volume, frequency, means of entry, source documents and their disposition, and security and privacy considerations. For output, provide file names, report formats, selection criteria for displays or outputs, types of output media, and disposition of products.

3.1.3 Perform Walkthroughs

Arrange one or more review walkthroughs to educate the attendees on the system/subsystem design and specifications. Include representatives from 1) other development and maintenance teams whose programs interface the system/subsystems being designed, 2) the system acceptance test team staff, 3) and user and sponsor organizations. Choose a walkthrough moderator to record action items and track them to completion. Incorporate feedback from the attendees into the design and specifications.

3.1.4 Document System/Subsystem Specifications

Prepare the System/Subsystem Specifications document in accordance with HUD SDM documentation standards (Handbook 2400.15).

Update the requirements traceability matrix to reflect the allocation of system objectives to system/subsystem design components.

Return to Title Page

3.2 DEVELOP DATABASE SPECIFICATION

Develop and document the detailed design for the system's databases. Describe the database management system and identify the database administrator. Include the following activities:

3.2.1 Identify the unique database names.

3.2.2 Identify any special instructions for database usage.

3.2.3 Reference all support software.

3.2.4 Identify database security considerations, sensitivities, and critical issues.

3.2.5 Identify the system administration and control personnel.

3.2.6 Identify the database management system configuration.

3.2.7 Describe the database schema.

3.2.8 Prepare the database specifications document.

Table 3-4 defines the roles and responsibilities of key personnel for developing database specifications.

Table 3-4. Roles and Responsibilities for Developing Database Specifications

Role	Responsibility
Project Manager	Coordinates development of database specifications. Identifies the organizational areas and individuals who will serve in the following capacities: database administrator, system administrator, system control officer, and security administrator. Coordinates capacity planning with Computer Services.
User	Supports database definition activities.
Developer	Prepares database specifications. Updates requirements matrix.
Project Data Administrator	Defines database design. Supports preparation of database specifications.
Quality Assurance	Reviews database specifications for adherence to standards.

3.2.1 Identify the Unique Database Names

Specify the code name, tag, or label by which each database table or file may be uniquely identified. Additional descriptive information also should be given, whether or not it is implied in the identification code, such as the system using the database, the database status, and a physical description of the database table or file.

3.2.2 Identify Any Special Instructions for Database Usage

Provide any special instructions to personnel who will contribute to the generation of the database or who may use it for testing or operational purposes, including criteria, procedures, and format for submitting data for entry into the database. Identify a data control organization.

3.2.3 Reference All Support Software

Describe all support software directly related to the database. Descriptions should include name, function, major operating characteristics, and machine run instructions for using the support software. Cite the support software documentation by title, number, and appropriate sections. Examples of support software are database management systems, query language, report writers, storage allocation software, database loading software programs, and file processing programs.

3.2.4 Identify Database Security Considerations, Sensitivities, and Critical Issues

Describe the security-related requirements of the database. Include such items as whether the database will contain sensitive data or data critical to the operation of the system. If sensitive or critical data will be part of the database, identify those measures put in place to safeguard the database, such as methods to guard against unauthorized entry into the database and to determine if unauthorized entry has been attempted, contingency plans if the database cannot be accessed, and steps to be taken to recover lost or damaged data.

3.2.5 Identify the System Administration and Control Personnel

Identify the organizational areas and individuals who will serve in the following capacities:

- ***Database administrator*:** The person responsible for the planning, definition, organization, protection, and efficiency of data and databases within HUD and specific to the project.
- ***System administrator*:** The person in the system owner organization within HUD who has responsibility for the administration of the system. This individual gives the final approval to a user’s request for access to the system.
- ***System control officer*:** The person within each HUD organization who reviews and approves access request by users within their organization. This person determines the type of access the user needs to perform his/her tasks.
- ***Security administrator*:** The person who controls the necessary activities that grant a user access to application systems based on access requests that have been approved by the system administrator.

3.2.6 Identify the Database Management System Configuration

Identify and describe the database management system (DBMS) configuration. Include such information as the operating system and storage device on which the DBMS is to be located, identification of the application software, all related computer programs that run the database, and the amount of online and offline storage required. If special hardware or software is required to maintain backup copies or archived files, identify these also.

3.2.7 Describe the Database Schema

Describe the database schema and related subschemas, including the names of all database-related application programs and data files, as well as the file sizes of each and data record layouts and descriptions.

Return to Title Page

3.2.8 Prepare the Database Specifications Document

The development team prepares the Database Specifications document in accordance with HUD SDM documentation standards (Handbook 2400.15).

3.3 DEVELOP PROGRAM SPECIFICATION

For each lowest-level function identified in the System/Subsystem Specifications, prepare a detailed program specification. Develop in detail and document the functions, timing requirements, interfaces, input and output reports, and the accuracy and validity requirements. Include the following activities:

- 3.3.1 Describe functions of the software units.
- 3.3.2 Describe accuracy and validity characteristics imposed on software units.
- 3.3.3 Describe timing requirements placed on software units.
- 3.3.4 Describe support software.
- 3.3.5 Describe interfaces with other application software.
- 3.3.6 Describe storage requirements.
- 3.3.7 Describe the degree of security required.
- 3.3.8 Describe the input and output data and output reports.
- 3.3.9 Define the data retention period.
- 3.3.10 Describe the logic of each software unit.
- 3.3.11 Document the program specifications.

Table 3-5 defines the roles and responsibilities of key personnel for developing program specifications.

Table 3-5. Roles and Responsibilities for Developing Program Specifications

Role	Responsibility
Project Manager	Coordinates development of Program Specifications. Coordinates capacity planning with Computer Services.
User	Reviews preparation of Program Specifications.
Developer	Prepares Program Specifications.
Quality Assurance	Reviews Program Specifications for adherence to standards.

3.3.1 Describe the Functions of the Software Units

Describe the functions that each program is expected to perform, encompassing a level of detail that will allow the program to be coded in software units. The functions should be taken directly from the corresponding sections of both the Functional Requirements Document and the System/Subsystem Specifications.

3.3.2 Describe Accuracy and Validity Characteristics Imposed on Software Units

Describe the data accuracy requirements and data validity requirements imposed on each software unit. Data accuracy may include identifying any mathematical calculations to be performed by a program. Data validation identifies the edit and error routines to be run on the data, such as ensuring that an all-numeric field contains only numbers and no alphabetic or special characters.

3.3.3 Describe Timing Requirements Placed on Software Units

Describe the timing requirements imposed on each software unit under varying conditions such as throughput time, response time to queries and updates of data files, response time of major program functions, sequential relationships of program functions and data flows, priorities imposed by types of input and change in modes of operation, timing requirements for the range of traffic load under varying operational conditions, and input/output transfer time required for disk and tape.

3.3.4 Describe Support Software

Provide a description of the support software with which each software unit must interact. Include support software, test software, and security software, if needed. Provide the correct name, description, and documentation references of each such software system, subsystem, and program. Included must be a reference to the language (compiler, assembler, program, query) and to the operating system to be used. If the operation of the software units to be developed is dependent on forthcoming changes to other software, identify and discuss the nature, status, and anticipated availability date of such changes.

3.3.5 Describe Interfaces With Other Application Software

Describe the interfaces with other application software, including those from other operational organizations. For each interface, include the following information: type of interface, such as operator control of a terminal or program interfaces with other programs; description of operational implications of data transfer, including security considerations; data transfer requirements to and from the subject program (including data content, sequence, timing, format, volume, and processing); current formats of interchanged data; interface procedures, including telecommunications considerations; interface equipment; and data conversion requirements.

3.3.6 Describe Storage Requirements

Describe the data storage requirements, including internal storage requirements, use of internal storage and auxiliary storage such as tape and disk, and the estimated quantity of storage required for each. Each program must consider the following types of information for the various storage media:

- ***Internal*:** Describe and illustrate the use of internal storage areas, including indexing and working areas. Briefly state the equipment constraints and design considerations that affect the use of internal storage.
- ***Device*:** List by device type all peripheral storage required and any constraints imposed on storage requirements by each storage device. State requirements for permanent and temporary storage, including overlays.

- ***Offline*:** Describe the form, media, and storage requirements of all offline storage.

3.3.7 Describe the Degree of Security Required

Determine the security classifications of the data used by the programs and the degree of security required for the algorithms. Determine if the data is always in a sensitive state. If the data becomes sensitive upon the occurrence of specific events, describe those events. Specify the operational environment that must exist when sensitive data of the system/subsystem are being processed.

3.3.8 Describe the Input and Output Data and Reports

Describe the input and output data and reports. For inputs, provide information about the characteristics of each input to the program such as file name, format and type of data; validation criteria; volume, frequency, and means of entry; source document and its disposition; and security and privacy considerations. For output, provide file name, report format, selection criteria for display or output, output media, and disposition of products.

3.3.9 Define the Data Retention Period

Determine the period of time the system/subsystem is required to retain its data and include this retention period in the program specifications for each of the data files used and/or created by the system.

3.3.10 Describe the Logic of Each Software Unit

Describe the logic of each software unit in the Program Specifications. Present the logical flow in graphical format showing the hierarchical arrangement of the software units and their interfaces. Supplement the graphic with pseudo-code and English language narrative explanations. Explain in detail the methods for identifying error conditions and the resulting actions of the program.

3.3.11 Prepare the Program Specifications Document

The development team prepares the Program Specifications document in accordance with HUD SDM documentation standards (Handbook 2400.15.)

Update the requirements traceability matrix to reflect the allocation of requirements to software units.

3.4 IDENTIFY HARDWARE AND SOFTWARE SUPPORT REQUIREMENTS

Identify all hardware and software support requirements. Determine these requirements based on the following three factors:

- Needs of the specific application, taking into account available equipment resources, available funding, and HUD's overall systems plans.
- Current technical capabilities available to HUD, including the availability of technically trained personnel and special training necessary to accommodate new hardware or software.

Return to Title Page

- Assessment of the stability of the technology in place at HUD, primarily the potential impacts of adding a system to current hardware and staff responsibilities.

After these factors are investigated, the logical conclusion may be to procure additional hardware and software to absorb all or part of the requirements of the system/subsystem under development. If so, update the Project Support Plan to reflect the hardware and software needed for the project. Include the following activities:

3.4.1 Identify hardware requirements.

3.4.2 Identify software requirements.

Table 3-6 defines the roles and responsibilities of key personnel for identifying computer hardware and software support requirements.

Table 3-6. Roles and Responsibilities for Identifying Computer Hardware/Software Support Requirements

Role	Responsibility
Project Manager	Coordinates system support activities.
User	Determines additional hardware and software support requirements.
Developer	Identifies hardware and software support requirements.
Computer Services	Assists with installation and checkout of development and test hardware and software.

3.4.1 Identify Hardware Requirements

Identify the hardware needed to fulfill all or any part of the requirements of the system/subsystem under development, such as workstations, operational system components, and communication hardware. For each item of hardware to be procured, identify the performance and functional requirements it must meet and the preferred vendor(s). Also include expected maintenance requirements if the additional equipment will cause a significant change in the current maintenance schedule.

Determine Workstation Hardware Requirements

Based on the detailed design specifications, Functional Requirements Document, and the Needs Statement, estimate the number of users to be assigned to the system/subsystem and determine the workstation hardware requirement for each. Include any development and testing and all related security requirements in this determination.

Determine Operational System Hardware Requirements

Estimate the impact of the proposed system’s requirements on existing operational hardware as well as any new requirements that result from the detailed system design. Include any additional hardware requirements that may result due to development, testing, maintenance, and related security requirements.

Determine Communication Hardware Requirements

Analyze the specifications outlined in the detailed technical design documentation. Perform an evaluation of the estimated traffic flow and network capability requirements. Based on this analysis, determine the communication hardware requirements of the proposed system. Include additional requirements that may occur due to the impact of the proposed system on existing communication networks and any additional requirements that may result due to additional impacts on security.

3.4.2 Identify Software Requirements

Identify software requirements, including system support software and communication software. For each item of software to be acquired, identify the functional and performance requirements to be met, preferred vendor(s), estimated additional online and offline storage required, and estimated system resources necessary to process the application.

Determine System Support Software Requirements

Based on the system/subsystem design, ascertain the functional and performance requirements for system support software. Estimate the additional online and offline storage requirements and any additional system resources that may be required. OIT should be consulted when making this determination.

Determine Communication Software Requirements

Based on the detailed system/subsystem design, ascertain the functional and performance requirements for system communications software. Estimate the potential impact these requirements will have on existing HUD communication systems. OIT should be consulted when making this determination.

3.5 CONTINUE ACQUISITION AND SUPPORT PLANNING ACTIVITIES

Those acquisition and support planning activities that were initiated during Define System are continued. These activities are related to the acquisition of hardware, software, and system support items required to continue with the project. At this point in the lifecycle, such tasks typically include continuing the acquisition process for the operating system hardware and environment; supporting software packages; vendor(s) services to provide and maintain these tools; and contractor support for the development, testing, and maintenance of the system. The following activities are included:

3.5.1 Continue system support activities.

3.5.2 Continue acquisition activities.

3.5.3 Update System Support Plan.

Table 3-7 defines the roles and responsibilities of key personnel for acquisition and support planning activities.

Table 3-7. Roles and Responsibilities for Acquisition and Support Planning Activities

Role	Responsibility
Project Manager	Coordinates system support activities with Operations.
User	Determines additional support requirements.
Developer	Identifies hardware and software support requirements.
Computer Services	Confirms and approves additional project support requirements. Provides updates to the System Support Plan.

3.5.1 Continue System Support Activities

Continue the system support activities initiated during the Define System phase to ensure that supporting organizations, such as OIT, are aware of the project's requirements and any revisions to those requirements and are currently (or are in the process of becoming) capable of meeting those requirements. Continue coordination to determine if the resources of other projects under development may be combined to acquire the necessary hardware or software.

Coordinate with Computer Services

Based on the platform and the requirements of the system, contact the appropriate Computer Services organizations to coordinate support activities to include the following areas:

- Hardware and software installation and checkout for each site
- Supply requirements
- Online and offline storage requirements
- Communications hardware and software
- Report routing

Ensure that all planned system support activities are consistent with the operations guidelines. Before beginning these activities, Operations must be given sufficient advanced notification to prepare the system environment and determine the impact on current operations.

3.5.2 Continue Acquisition Activities

Continue the tasks begun during the Define System phase that are necessary to acquire the hardware and software for the development and operation of the system. If necessary, document in the System Support Plan any changes in the strategy.

Continue Software Acquisition Activities

Continue the software acquisition activities initiated during the Define System phase to ensure the ongoing review and evaluation of potential supporting software packages (such as database management system, data entry system, communication software, security software). Ensure that the necessary paperwork for acquisition of the required software is being prepared and submitted on schedule.

Return to Title Page

Continue Hardware Acquisition Activities

Continue hardware acquisition activities initiated during the Define System phase, including the review, evaluation, and acquisition of operating system hardware, communications hardware, online terminals, printers, security devices, or additional disk storage. Ensure that the necessary paperwork for acquisition of the required hardware is being prepared and submitted on schedule.

3.5.3 Update System Support Plan

Update the System Support Plan to include any additional information derived during the Design System phase.

3.6 DEVELOP SYSTEM TESTING STRATEGY

Document the system testing strategy in the Verification, Validation, and Test (VV&T) Plan and provide for acceptance testing of all components of the system, including detailed requirements for all tests, testing methods and tools, and test evaluation criteria. Include the following activities:

3.6.1 Identify test evaluation criteria.

3.6.2 Identify test resources.

3.6.3 Describe testing methods and tools.

3.6.4 Prepare preliminary VV&T plan.

Table 3-8 defines the roles and responsibilities of key personnel for developing the system testing strategy.

Table 3-8. Roles and Responsibilities for Developing System Testing Strategy

Role	Responsibility
Project Manager	Coordinates development of system testing strategy.
User	Reviews and confirms system testing strategy.
Developer	Provides input related to the development testing strategy for the VV&T Plan.
System Acceptance Test Team	Develops system testing strategy. Identifies test resource and test evaluation criteria. Describes testing methods and tools. Prepares preliminary VV&T plan. Updates requirements matrix.
Computer Services	Provides input to system testing strategy.
Quality Assurance	Reviews VV&T Plan for adherence to standards.

3.6.1 Identify Test Evaluation Criteria

Determine the specific criteria that each segment of the system/subsystem must meet. Such criteria are described by the user of the system/subsystem and typically are a mix of functional and performance requirements, such as processing data within a certain time frame, producing a report, or responding to an online query within a certain amount of time. The majority of the tests to be included in the VV&T Plan should be designed to prove the system's ability to meet the user's acceptance criteria.

Determine User System Acceptance Criteria.

Determine the minimum function and performance criteria that must be met for the system to be accepted as "fit for use" by the user or sponsoring organization.

Obtain Concurrence(s) on Acceptance Criteria

Obtain the approval of the user or sponsoring organization on the minimum criteria the system must meet for it to be accepted as "fit for use" by their organization.

3.6.2 Identify Test Resources

Identify those items that will be needed during the execution of each system test. Prepare a testing schedule to reflect unit, integration, and system acceptance testing activities. Provide the following information:

- Software requirements
- Deliverable materials
- Security considerations
- Site-supplied materials
- Personnel and equipment requirements

Prepare a Testing Schedule

Prepare a testing schedule to reflect the unit, integration, and system acceptance tests and the time duration of each. This schedule should reflect the personnel involved in the test effort and the site location. In the test schedule, reflect the following information:

- Documentation review
- Data preparation
- Test execution
- Output review
- System certification
- System release
- Return of test site to pretest condition

Return to Title Page

Identify Software Requirements

Identify any software requirements that will be needed for the tests, including software tools or utilities, such as tape compares. Include version numbers and media type for all required software.

Identify Deliverable Materials

List all deliverable materials, including technical and documentation deliverables that will be needed for the tests. Version and release numbers should be included as well as the names of any supporting off-the-shelf software requirements.

Identify Any Security Considerations

Prepare a list of requirements necessary to ensure the integrity of the testing procedures, data, and site. Any special security considerations (e.g., passwords, classifications, security or monitoring software, or computer room badges) should be described in detail.

Identify Site-Supplied Materials

Describe any materials required to perform the test that need to be supplied at the test site. These materials could include desks, chairs, special equipment, and office supplies, as necessary.

Identify Personnel Requirements

A listing of the personnel necessary to perform the test should be provided. For each of the personnel, this listing should provide the following information:

- Name, title, current organization, grade (if known), and level of security background investigation
- Description of the required tasks to be performed
- Geographical location of the work to be performed
- Time required (dates needed)
- Whether the requirement is full time, part time, or as needed
- Any special skills required (i.e., programming language, machine familiarity)

Identify Equipment Requirements

Describe the equipment needed to perform the test. Include the type of equipment, the amount, the location, and the time periods needed.

3.6.3 Describe Testing Methods and Tools

Describe the testing methods and tools. First, describe the testing method to be used during the system test. Then, identify the testing tools to be used during the preparation for and execution of the test. Finally, describe the specific tasks and activities to be accomplished during the designated testing period.

Identify Testing Methods and Tools

Analyze the detailed design specifications to determine the appropriate methods to employ when performing the unit, integration, and system tests. In this analysis, include a reference to all tools that may be required in the performance of the test. Cross-reference the methodologies and tools selected with the design specifications to ensure that all areas of the system are thoroughly tested before their operational release.

Describe Testing Tasks and Activities

Based on the determinations made in the analysis of the detailed design specifications, prepare descriptions of the necessary tasks and activities that will be performed during the described tests. Include the methodology and tools to be employed during testing. Prepare a cross-reference to the system's detailed design documents to ensure that all tasks and activities necessary to perform a thorough test of the system are described.

3.6.4 Prepare Preliminary VV&T Plan

Document all information in the VV&T Plan in accordance with HUD SDM documentation standards (Handbook 2400.15).

Update the requirements traceability matrix to reflect the allocation of requirements to acceptance criteria.

Describe the overall activities involved with testing and the schedule for performing the activities. For each level of testing (unit, integration, and system) document the following information: organizations involved, hardware and software required for testing, location of tests, test data development, test scenarios, evaluation criteria, procedures for testing, and certification procedures.

3.7 DETERMINE TRAINING APPROACH

Develop the training approach to be used on the project. Tasks include identifying the training methods and tools, preparing a preliminary training schedule, developing an appropriate curriculum, and preparing the training plan. Include the following activities:

3.7.1 Identify training methods, techniques, and tools.

3.7.2 Identify training required for revised office procedures.

3.7.3 Prepare preliminary training schedule.

3.7.4 Develop curriculum.

3.7.5 Prepare training plan.

Table 3-9 defines the roles and responsibilities of key personnel for developing the training plan.

Table 3-9. Roles and Responsibilities for Developing Training Plan

Role	Responsibility
Project Manager	Coordinates development of training plan.
User	Confirms training plan and materials.
Developer	Identifies training methods, techniques, and tools. Develops curriculum. Prepares training plan.
Computer Services	Provides input to training approach.
Quality Assurance	Reviews training plan and materials for adherence to standards.

3.7.1 Identify Training Methods, Techniques, and Tools

Describe the training methods to be used for the project (e.g., whether the HUD staff will conduct the training or whether the services of a vendor will be acquired). Determine the training techniques to be used (e.g., computer-based instruction, self-paced written manual, peer training, hands-on practical sessions, classroom lectures, or any combination of these). Identify the tools needed for the training, such as online terminal, training manual, classroom, and computer center.

3.7.2 Identify Training Required for Revised Office Procedures

Identify the training needs for the user's staff if the advent of the system under development will change the procedures of the user's office in any way. Because user acceptance of the system is paramount, consider all changes expected to the user's current processing, no matter how small. The project sponsor is the best qualified to identify projected changes to office procedures; however, cooperation from the developer's staff as well as the ADP Security staff is necessary because they are most knowledgeable about the new system operation and security requirements.

3.7.3 Prepare Preliminary Training Schedule

Prepare a training schedule to include the following information:

- Planned training dates
- Names of students
- Names of instructors
- Location of the sessions

This schedule should be as comprehensive as possible; however, the schedule may be revised at later points in the project's lifecycle.

3.7.4 Develop Curriculum

Determine the job classifications of the individuals that will need to be trained on the use of the system. Using the system/subsystem specifications and other system development documents, determine the system functions that each job class must be familiar with to enable them to successfully interface with the system. Using this information, determine the different courses and the materials that must be presented in each course.

3.7.5 Prepare Training Plan

Document all information regarding the training strategy and approach in the Training Plan, in accordance with HUD SDM documentation standards (Handbook 2400.15). Include the schedule of training courses, a description of each course, and list of the materials required.

3.8 UPDATE SYSTEM AUDIT STRATEGY

Notify the IG staff about any work products completed during the Design System phase and informed of any changes that may have an impact on internal data processing controls as they were described in the Design System phase documentation. Activities include coordinating with IG staff, providing input to IG staff, and updating the Internal Audit Plan.

Table 3-10 defines the roles and responsibilities of key personnel for updating the system audit strategy in the Internal Audit Plan.

Table 3-10. Roles and Responsibilities for Updating System Audit Strategy

Role	Responsibility
Project Manager	Provides project information to IG staff, as requested.
IG Staff	Coordinates with project manager, as necessary. Updates internal audit plan.

Activities include the following:

- ***Coordinate with IG Staff.*** Schedule meeting(s) with the Inspector General's staff to inform them of any developments that occurred during the Design System phase. Determine the scope of the information required from the project manager in order for the IG staff to perform the necessary updates to the Internal Audit Plan.
- ***Provide input to IG staff.*** Provide the IG staff with the necessary information.
- ***Update Internal Audit Plan.*** Update the Internal Audit Plan to include information provided by the project manager.

3.9 UPDATE SYSTEM DECISION PAPER

Update the System Decision Paper to include a summary of the progress made during the Define System phase and the schedule of events for the Build System phase activities. Include the current status of the project. List in detail any changes to the original plan for the development of the project and the parties responsible for approving those changes. The updated System Decision Paper is then reviewed and approved by the appropriate level of management. The following activities are included:

3.9.1 Summarize progress of system.

3.9.2 Identify changes.

3.9.3 Summarize schedule of events.

3.9.4 Summarize status.

3.9.5 Document results

Table 3-11 defines the roles and responsibilities of key personnel for updating the System Decision Paper.

Table 3-11. Roles and Responsibilities for Updating System Decision Paper

Role	Responsibility
Project Manager	Updates System Decision Paper, as necessary.
Developer(s)	Provides inputs needed to update System Decision Paper.
Quality Assurance	Reviews System Decision Paper for adherence to standards.

3.9.1 Summarize Progress of System

Summarize the progress of the project development through the Design System phase, and update the System Decision Paper accordingly.

3.9.2 Identify Changes

Document any changes to the system development milestones, and schedule any modifications made to the project strategy. Update the System Decision Paper accordingly.

Describe Changes in Milestones, Schedules, and Tasks

Provide a description of any Design System phase occurrences that may have had an impact on projected milestones, schedules, and tasks of the development effort. Provide a brief explanation of the nature of these impacts, and describe any changes to the milestones or schedule that may occur as a result of these impacts.

Describe Modifications to Project Strategy

Provide a description of any modifications to the project strategy that have occurred during the Design System phase. Provide a brief explanation of these modifications, indicating the reasons for the changes as well as any anticipated impacts.

3.9.3 Summarize Schedule of Events

Develop a brief summary of events that took place during this stage of development and a description of the remaining schedule of development activities. Show the impact these events have had on the original schedule and any anticipated impacts on future development activities.

3.9.4 Summarize Status

Provide a status report of the development of the system. Include in the status report the percentage of the system that has been developed and successfully passed integration testing and its readiness for acceptance testing, as reported by the integration test team in the test analysis report. Indicate the components or subsystem of the system that may be causing delays or problems.

Return to Title Page

3.9.5 Document Results

Document all information for this phase in the System Decision Paper in accordance with HUD SDM documentation standards (Handbook 2400.15).

3.10 REFINE PROJECT PLAN

The Project Manager controls the project by monitoring phase activities, taking corrective action where necessary, and refining the project plan to account for changes attributable to actions taken in the current phase and new information for upcoming phases. The Project Manager also reviews the quality process planned by users, developers, testers, and quality assurance for the next phase. Activities include the following:

3.10.1 Update Project Plan.

3.10.2 Review planned quality process.

Table 3-12 defines the roles and responsibilities of key personnel for refining the Project Plan.

Table 3-12. Roles and Responsibilities for Refining the Project Plan

Role	Responsibility
Project Sponsor	Approves changes to project schedule. Approves changes to Project Plan.
Project Manager	Updates Project Plan to include actual costs incurred and specific activities accomplished for the Design System phase; revises project schedule, plans, strategies, resources, and requirements for subsequent phases, as required. Reviews, for approval, the quality process planned by users and developers for the next phase.
User	Provides updated input to the project manager on required Design System phase activities. Works with developer to review and update the quality process planned for the next phase. Attends the review of the Project Plan to provide technical input to the review process, as required.
Developer	Provides updated input to the project manager on required Design System phase activities. Works with user to review and update the quality process planned for the next phase.
Configuration Management	Provides updated input to the Project Plan.
Quality Assurance	Provides updated input to the Project Plan.

3.10.1 Update Project Plan

Update the Project Plan with cost, schedule, and budget data for the current phase (Design System) at the level of detail necessary to reflect the status of the project. Review the plan and determine if any activities described for the next phase (Build System) and subsequent ones are affected by the completion status of the current phase activities. Adjust the schedules and resource requirements for activities in the next phase, if necessary, and assign starting and ending dates for activities that have been affected. Take into account that the starting date for some activities may be dependent on the completion of other activities. Update milestones, schedules, and resource requirements for the remainder of the project.

3.10.2 Review Planned Quality Process

Users, developers, testers, and QA work closely together to determine the process to be used to build the product with the desired quality. Activities include tailoring the planned quality process for the next phase and identifying the standards and procedures to be used. The project manager reviews the quality process for approval and execution in the next phase.

3.11 REVIEW AND APPROVE DOCUMENTS AND DELIVERABLES

The review and approval of documents and deliverables is an iterative process as each phase provides more definition and products are revised. Additionally, the approval process offers senior management the capability to monitor the activities of the project. The following activities are included:

3.11.1 Review revised products from prior phases, as required.

3.11.2 Submit Design System phase deliverables for review and approval.

3.11.3 Obtain approval of project documents and deliverables.

Table 3-13 defines the roles and responsibilities of key personnel for review and approval of documents and deliverables.

Table 3-13. Roles and Responsibilities for Review and Approval of Deliverables and Documents

Role	Responsibility
Project Sponsor	Participates in deliverable reviews (optional). Approves Project Plan revisions.
Project Manager	Determines required review level and schedules reviews. Coordinates review and approval of Design System deliverables. Determines needed changes to prior lifecycle phase products and ensures revision in accordance with the Project Plan. Attends reviews and presents deliverables. Obtains concurrences and approvals for deliverables and associated management summaries. Coordinates review of all deliverables.
Development Team	Assists in preparation of deliverable reviews. Attends deliverable reviews and provides input as necessary. Prepares deliverable documents. Provides technical expertise during review process. Revises prior lifecycle products. Updates deliverables to include any recommendations received from QA review.
ADP Security	Reviews deliverables to ensure incorporation of security requirements. Reviews prior lifecycle products to ensure that security standards and guidelines have been addressed.
System Acceptance Test Team	Reviews Project Plan.
Project Database Administrator	Reviews Project Plan and Database Specifications.
Computer Services	Participates in QA review of all documents prepared and revised during Design System phase.
Configuration Management	Assists project manager with review preparation.
Quality Assurance	Ensures all review procedures are followed, as required. Reviews Design System phase deliverables to ensure they meet all applicable HUD and project standards.

3.11.1 Review Revised Documents from Prior Phases as Required

Review the activities performed during the Design System phase to determine if they have had an impact on any documents produced during the previous phases. If changes are required to prior lifecycle phase documents, update those documents to reflect current project developments. The revised documents are reviewed by project personnel similarly to reviews for the current phase documents to ensure that the changes are within the scope of the project's requirements and are in compliance with HUD and project standards and procedures. Prepare a management summary for each revised document that summarizes the essential revisions. Submit the revised products to the appropriate review board for approval. Affected product change records will be filed with CM for appropriate version control updates. Use the appropriate document review checklist in Appendix E to aid in the document reviews.

3.11.2 Conduct Review for Design System Phase Documents and Deliverables

Conduct a project review with project personnel and system stakeholders to ensure that the project documents and deliverables for the Design System phase include the necessary level of detail, fulfill the system's requirements, and meet the appropriate HUD and project standards and guidelines. Use the appropriate document review checklist in Appendix E for Design System phase deliverables to aid in the document reviews.

A minimum of 10 working days before the scheduled review, notify the personnel required to attend and provide each with a copy of the product for their pre-review. Discuss all comments or objections that are raised during the review, and reach a consensus on one of the following before the review session terminates:

- The document is correct and complete, as is, without any further changes.
- Additional changes need to be made but are minor and do not require further review. In this case, the updates should be made and change pages should be distributed by an agreed-upon date.
- Required changes will have a major impact on the plan. A second review must be scheduled. The changes must be incorporated and the resulting change pages distributed at least 10 working days before the second review.

Prepare a management summary for each document that includes the essential data collected in the document, lists conclusions that may be drawn from the document, and describes the potential impact on the project, if applicable. Submit the documents to the appropriate review board for approval. Documents for approval include the following:

- System/Subsystem Specifications
- Program Specifications
- Database Specifications
- Training Plan
- Preliminary System Test Plan
- System Support Plan (updated)
- System Decision Paper (updated)
- Project Plan (updated)

3.11.3 Obtain Approval of Project Documents and Deliverables

Present the project documents and deliverables to the chairperson of the appropriate review board for approval at least 10 days before the scheduled decision date. Include the management summary information, approval (sign-off) record, the System Decision Paper, and any other required or requested information. The review board chairperson coordinates review board comments, recommendations, and approval signature(s), and returns the approval record to the project manager. Recommendations on the approval record are addressed by the project and the document should be resubmitted to the board if requested. Approval will be assumed if there is no response, or the response is "no comment." Project approval records are maintained by the project's configuration management (CM) function; a copy is inserted into the central project library.

The project proceeds to the next phase after all project documents and deliverables are approved.

3.12 PERFORM CHANGE CONTROL ACTIVITIES

Any products developed during the Design System phase are baselined and subjected to version control. Products baselined during prior activities are assigned new version control numbers when they undergo change (e.g., an update or rewrite). Configuration management reports are provided to the project manager as requested or required.

Table 3-14 defines the roles and responsibilities of key personnel for performing configuration change control activities for approved baselined products.

Table 3-14. Roles and Responsibilities for Performing Change Control Activities

Role	Responsibility
Project Manager	Approves controlled products for distribution.
Configuration Management	Performs change control for new and revised products. Provides status reports to project management as requested or required.
Quality Assurance	Audits products to ensure that only approved changes are addressed.

The Configuration Management function prepares the baseline for configuration control. The baseline includes any new technical and document deliverables that will comprise the project configuration baseline. For the baselined items, Configuration Management performs the following activities:

- ***Verify changes made to product.*** Review the updated product to ensure that the changes have been made as described in supporting documentation. Supporting documentation may be comments received from document reviews. Supporting documentation for software may be software change requests or discrepancy reports generated during testing.
- ***Assign version number*.** The version number must follow conventions established by HUD and enable the project's CM to monitor updates to the product and assist in its distribution. For software programs, increment the version number each time a change is made to the software to correct deficiencies found during testing.
- ***Store approved version in central library.*** After each baselined product has been completed and approved, according to HUD procedures, store the approved version in the project’s central library. The project's CM function controls access to the library. For software, the library can be a subdirectory or dataset where all baselined software configuration items will be stored.
- ***Record product information in inventory log.*** Maintain an inventory log that includes the title of the product, release date, version number, name and version or model numbers of the software and hardware used in the development of the product, name of the organization responsible for development of the product (usually the sponsoring organization), and the product distribution list.

Return to Title Page

- ***Distribute copies of products as required.*** The project's CM function distributes copies of the products according to a distribution list maintained as part of the inventory log information, with distribution based on need and the security level of the product.
- ***Archive old versions of products.*** Archive and retain outdated versions of all products for the required period of time, in keeping with HUD standards and guidelines.

Return to Title Page

Section 4. Build System

4.0 Build System

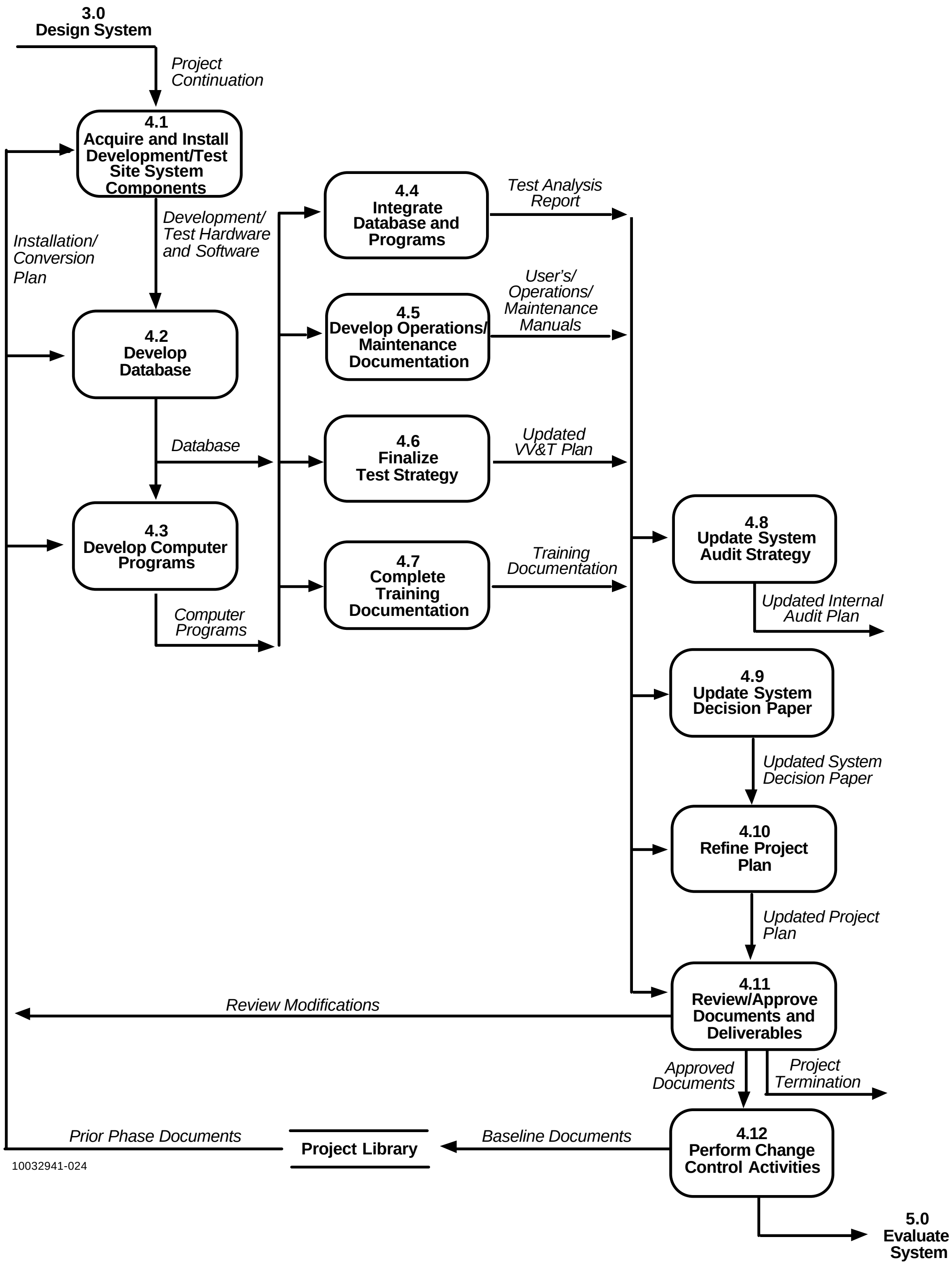

Figure 4-1. Process Flow for Build System Phase

4.0 BUILD SYSTEM

Purpose

The Build System phase of system development is the period in which developers take the detailed logical information provided in the Design System phase, transform it into machine-executable form, and ensure that all of the individual components of the system function correctly and interface properly with other components within the system. Figure 4-1 highlights the process flow for the Build System phase.

Overview

After the system design is completed and approved in the Design System phase, it is used as a blueprint to develop the database and computer programs that will be used by the target hardware and in conjunction with any specified support software to automate the system. To aid in this effort, any additional hardware or support software required to develop, test, or operate the system is procured and installed.

During this phase of system development, supporting organizations assist the project team in performing the following activities:

- Compile and record in system documentation (e.g., user's, operations, and maintenance manuals) information needed to use, operate, and maintain the system.
- Finalize and document in the Validation, Verification, and Test (VV&T) Plan the strategy for testing the system in the Evaluate System phase.
- Finalize in the Training Plan the strategy for training staff that will use and operate the system or will oversee users of the system.
- Provide staff from other affected HUD organizations, including the Inspector General’s Office, with updates on project status and any other requested project information.
- Maintain the Project Plan. Document project expenses for this phase and any new or revised estimates for subsequent phases to reflect cost and schedule for completing the project.
- Track all hardware and software, whether developed or procured, as well as all documentation, using the project's CM function.
- Revise and review prior products to reflect information learned during the Build System phase and review those products.
- Obtain approval of products produced during the Build System phase through project staff reviews and appropriate HUD review boards. Table 4-1 lists the functions and products resulting from the Build System phase.
- Use information gathered during this phase to update the System Decision Paper. Present the paper to the appropriate review board at the end of this phase. Projects that receive review board approval will continue on to the Evaluate System phase of the development lifecycle.

Return to Title Page

Table 4-1. Build System Phase Functions and Products

Build System Functions		Products
4.1	Acquire and Install Development and Test Site System Components	Installation and Conversion Plan (Preliminary)
4.2	Develop Database	Database
4.3	Develop Computer Programs	Computer programs
4.4	Integrate Database and Programs	Test Analysis Report
4.5	Develop Operations and Maintenance Manuals	User's, Operations, and Maintenance manuals
4.6	Finalize Test Strategy	VV&T Plan
4.7	Complete Training Documentation	Training Plan
4.8	Update System Audit Strategy	Internal Audit Plan (updated)
4.9	Update System Decision Paper	System Decision Paper (updated)
4.10	Refine Project Plan	Project Plan (updated)
4.11	Review and Approve Documents and Deliverables	New and revised products Management summary[1]
4.12	Perform Change Control Activities	Change control records

[1] A management summary is prepared for each product produced or revised during the Build System phase. This one-page summary includes a summary of the essential data collected in a document product, conclusions that may be drawn from the document, and potential impacts on the project, if applicable.

Standards and Guidelines

Development of the Build System phase products should comply with HUD SDM documentation standards (Handbook 2400.15) and project guidelines.

Roles and Responsibilities

Throughout the Build System phase of development, key personnel are required to perform various activities. Table 4-2 identifies the types of personnel required and the activities for which they are responsible.

Table 4-2. Roles and Responsibilities for Build System Phase (1 of 3)

Role	Responsibility
Project Sponsor	Approves changes to the project schedule. Approves changes to the Project Plan. Participates in reviews of deliverables.
Project Manager	Continues acquisition support activities. Coordinates installation of development and test hardware and software. Determines needed changes to documents of an earlier phase, and ensures revision in accordance with the overall Project Plan.

Table 4-2. Roles and Responsibilities for Build System Phase (2 of 3)

Role	Responsibility
Project Manager (Cont'd)	Monitors technical activities. Obtains appropriate concurrences and approvals for Build System documentation deliverables and associated management summaries. Provides project information to Inspector General's staff, as required. Determines required review level and schedules reviews. Determines needed changes to documents of prior lifecycle phases. Obtains appropriate concurrences and approvals of updated documentation. Attends reviews and presents deliverables. Updates Project Plan to include actual costs incurred and specific activities accomplished for the Build System phase; and revises project schedule, plans, strategies, resources, and requirements for subsequent phases, as required. Reviews, for approval, the quality process planned by users and developers for the next phase. Updates System Decision Paper, as necessary.
Project Development Team (User and Developer)	Identifies additional hardware and support software requirements. Coordinates development and test site hardware and software checkout and verification. Physically implements the database. Tests the database and evaluates results. Codes and unit-tests computer programs. Prepares unit and integration test plans. Integrates and tests computer programs. Updates requirements matrix. Prepares user's, operations, and maintenance manuals. Finalizes Training Plan. Reviews test strategy and VV&T Plan. Prepares preliminary Installation and Conversion Plan. Revises products of prior lifecycle phase. Assists with preparation of Build System phase deliverables for review. Provides technical expertise during the review process. Provides input for System Decision Paper updates. Provides input to project manager on remaining phase activities. Reviews and updates the quality process planned for the next phase.

Return to Title Page

Table 4-2. Roles and Responsibilities for Build System Phase (3 of 3)

Role	Responsibility
Computer Services	Assists with installation and checkout of development and test system hardware and software. Participates in database implementation. Participates in review of applicable documents and deliverables. Revises capacity planning estimates. Reviews the user's, operations, and maintenance manuals and Installation and Conversion Plan. Provides input to the system test strategy.
ADP Security	Reviews Build System phase deliverables to ensure that all necessary security requirements have been incorporated. Reviews revised products to ensure that the necessary security standards and guidelines have been addressed.
IG Staff	Coordinates with project manager, as necessary. Updates the Internal Audit Plan.
Quality Assurance	Reviews revised documents to ensure they meet applicable HUD and project standards and guidelines. Reviews Build System phase deliverables to ensure they meet applicable HUD and project standards and guidelines. Ensures that all review procedures are followed, as required. Monitors Build System phase activities for conformance to HUD and project standards and guidelines. Coaches users and developers in the planning for quality in the next phase.
Configuration Management	Performs change control for new products. Performs version control for revised products. Maintains changes to test data. Provides status reports to project management, as requested or required. Establishes controlled environment for development software test. Assists the project manager with preparation of Build System deliverables for review.
Project Data Administrator	Supports database implementation and test. Participates in reviews of deliverables and documents. Assesses application performance.
Project Database Administrator	Reviews database implementation and test results. Validates database design modifications.
System Acceptance Test Team	Finalizes system test strategy. Finalizes VV&T Plan Supports development and test site hardware and software checkout and verification.

Return to Title Page

4.1 ACQUIRE AND INSTALL DEVELOPMENT AND TEST SITE SYSTEM COMPONENTS

Procure, install, and verify all hardware and software required for coding, physical implementation of the database, and testing the system at all locations where these activities are to take place. Include the following activities:

4.1.1 Acquire system hardware.

4.1.2 Acquire system software.

4.1.3 Integrate and install system hardware and software at development and test site.

Table 4-3 defines the roles and responsibilities of key personnel for acquiring and installing system components.

Table 4-3. Roles and Responsibilities for Acquiring and Installing System Components

Role	Responsibility
Project Manager	Coordinates installation of development and test hardware and software.
Developer	Identifies additional hardware and support software requirements. Coordinates development and test site hardware and software checkout and verification.
Computer Services	Assists with installation and checkout of development and test hardware and software.

4.1.1 Acquire System Hardware

This activity began during the Define System phase, when the procurement process was started, to allow for timely delivery of hardware. During this activity, the project manager ensures the timely arrival of the correct hardware as specified as part of the system requirements and recorded in the system support plan.

Execute the final steps in the procurement of any additional hardware at this point of system development to ensure that the hardware will be ready for system development and test personnel. Select hardware in accordance with HUD's hardware platform standard.

Acquire Workstation Hardware

Procure all hardware that will be used as the input and output devices by the system users and operators. Workstation hardware may include terminals, microcomputers (multifunction workstations), portable workstations, scanning devices, pointing devices, remote printers available for users' access, and display and security hardware.

Acquire Operational System Hardware

Procure any additional operational system hardware that will be used as the central or front-end processing unit(s) and other equipment central to the operation of the system. Additional system hardware may include LAN servers, mini- or mainframe computers, Direct Access Storage Device (DASD) equipment, tape drives, security hardware, and central printing equipment.

Acquire Communication Hardware

Procure any hardware not presently available at HUD to provide connectivity between remote workstations and the Central Processing Unit (CPU) or other computer systems. Communication hardware may include switching equipment, cabling, multiplexers, modems, gateways, protocol converters, communications boards, and security hardware.

4.1.2 Acquire System Software

This activity began during the Define System phase, when the procurement process was started, to allow for the timely delivery of the software. During this activity, the project manager ensures the timely arrival of the correct software, and the correct version of the software, targeted for operation on the existing or newly acquired hardware.

Execute the final steps in the procurement of any additional software at this point of system development to ensure that the software will be ready for system development and test personnel. Select all software selected in accordance with HUD's software and security standards.

Acquire System Support Software

Procure any commercially available system support software not presently available at HUD to provide support to the system under development. System software may include DBMS, compilers, database query software, debuggers, security software, operating system software, and test support software.

Acquire Communication Software

Procure any software not presently available at HUD to provide connectivity between remote workstations and the CPU(s) or other computer systems. Communication software may include Network Operating Systems (NOSs), security software, gateway software, and protocol support and conversion software.

4.1.3 Integrate and Install System Hardware and Software at Development and Test Site

After the system hardware and software have been procured and any modifications to the physical facilities are completed as outlined in the system support plan, install the procured hardware and software and verify each component individually for correct operation. Thoroughly test security software and hardware to ensure a safe operating environment. After the hardware is procured and tested, install and test the procured software to ensure correct operation with the integrated hardware environment. Coordinate all hardware and software installations and perform a checkout with the operations organization appropriate for the system platform.

4.2 DEVELOP DATABASE

After the development and test system hardware and software have been installed and verified, and before the application programs have been developed, define and verify the structure of the database specific to the system being developed, in accordance with the DBMS. Test the database using actual data developed for the test. Perform a formal review of the database with participation from all organizations that have an interest in the system under development, data administration, or the operation of the system and database. Update the database specifications developed during the Define System phase to reflect any modifications to the structure of the database necessary to resolve issues found during database testing and review. Include the following activities:

4.2.1 Build database.

4.2.2 Test database.

4.2.3 Update database documentation.

Table 4-4 defines the roles and responsibilities of key personnel for developing the database.

Table 4-4. Roles and Responsibilities for Implementing Database

Role	Responsibility
Project Manager	Monitors database activities.
Developer	Physically implements database. Tests database and validates test results.
Computer Services	Participates in database implementation.
Project Data Administrator	Supports database implementation and test.
Project Database Administrator	Reviews database implementation and test results. Validates database design modifications.
Quality Assurance	Monitors activities for conformance to HUD and project standards and guidelines.

4.2.1 Build Database

Using the DBMS design as described in the database specifications, implement the design in the selected DBMS. This effort entails creating the database on the hardware in the physical environment, performing a review to check for errors, and then making any necessary modifications to the database design.

Create and Load Data Into Database

Using the DBMS, and any software supported by the DBMS, build the physical database structure in which the data will reside. This effort includes the development of all datasets and files containing the correct record structure, associated keys, and the development of all index and validation files.

Upon establishing the database, load test data to be used during testing of the database and development of application programs. The data may be actual data from existing automated systems, with modifications to test for error checking, or may be developed specifically for the purposes of testing.

Review Database

Hold a review of the physically implemented database to ensure that the database is implemented as designed in the database specification. Coordinate the review of the database with the operations organization appropriate to the platform of the system and with HUD's Data Administration Branch to ensure that data standards and guidelines are followed. This review will focus on ensuring the following:

- Records defined in the database specifications exist and all fields are accounted for.

- Required index files are present. Primary and secondary keys are present.
- Files in the database are sized correctly.
- Edit or validation tables have been defined.
- Aliases or synonyms are present.
- All applicable database and data standards and guidelines have been followed.

Refine Database as Required

Modify and reload the database structure to correct any deficiencies found during review of the database. Reload test data and, if necessary, perform a review of the modifications to ensure correctness.

4.2.2 Test Database

Test the physically implemented database and verify it before any live data are added. Testing should be completed before development of application programs that access the database because changes to any part of the database will affect programs using that portion of the database. The test to verify that the design will operate as expected focuses on such items as the following:

- Security of the data
- Database performance
- Considerations for operating the database on the target platform
- File and database sizing
- Data backup and recovery

Create Database Tests

Develop a plan for testing the database before test execution, and identify the tests that must be performed to execute the plan. Develop and document the tests and the data prepared by adding new data to the database or modifying existing data. The test documentation can be in any format within HUD guidelines that is agreed upon by the project team, but it generally will contain the following information:

- Description of the test
- Steps to be taken to execute the test
- Data to be used to execute the test
- Expected results

Perform a QA walkthrough of the tests before test execution to ensure that all necessary testing will be performed.

Execute Database Tests and Record Results

Once the database tests have been developed and approved, execute them following the steps described in the test documentation. Note any deviations from the original test and record them in the test documentation.

After the test is executed, record the results as part of the test documentation along with any supporting data, reports from the database, or reports generated by the system software during execution of the test.

Evaluate Test Results

Compare and validate the actual results from the execution of the tests to the expected results. Make decisions on the acceptability of the deviations. Document acceptable deviations as part of the test documentation. For unacceptable deviations requiring modifications to the database, determine how to correct the error or problem.

Correct Database Errors

Modify the database to correct errors. Make modifications to correct performance issues. Additional security levels or data encryption may be imposed on certain data to correct security deficiencies. Resize files or the entire database, if necessary, to correct space deficiencies.

4.2.3 Update Database Documentation

After the database has been tested and any necessary modifications made, update the database specifications document to reflect those changes. If changes are significant, review the changes with the HUD Data Administration Branch for adherence to HUD data standards.

4.3 DEVELOP COMPUTER PROGRAMS

During this portion of the Build System phase, the system's computer program design is transformed into machine operating instructions. Along with the development of the programs, the effort to test and make any necessary program modifications, at both the unit and integration levels, is also performed. Record and store all test results in the project library. Document all modifications to program designs necessary as a result of errors found during testing by updating the appropriate Program Specifications or the System/Subsystem Specifications or both. Include the following activities:

4.3.1 Prepare structure charts.

4.3.2 Develop programs.

4.3.3 Unit test programs.

Table 4-5 defines the roles and responsibilities of key personnel for developing computer programs.

Table 4-5. Roles and Responsibilities for Developing Computer Programs

Role	Responsibility
Project Manager	Monitors implementation of computer programs.
Developer	Codes and unit-tests computer programs. Prepares unit test plans. Updates requirements matrix.
Configuration Management	Establishes controlled environment for development software test.
Quality Assurance	Monitors activities for conformance to HUD and project standards and guidelines.

4.3.1 Prepare Structure Charts

Using the system's Program Specifications as a guide, develop the program structure chart(s), depicting program modules and hierarchical relationships between modules. A module can consist of one or more related software units. If a CASE tool is being used, develop the program structure using appropriate functions of the tool.

Identify Modules

To ease programming and eventually program maintenance, break the program down into manageable portions or modules. Each module should perform one unique function in the program, use all data that is passed to it, and have one entry and one exit point only.

Hierarchically Arrange Modules

Once all the modules that will perform the functions of the program have been identified, determine the data or flags that must be passed between modules and the sequence in which the modules must be executed. Additional modules may be needed to control the sequence in which the lower-level modules will be executed. More than one layer of control modules may be necessary. As modules that control the sequence of execution are added, the hierarchy of the program is refined.

Identify Data Access Strategy

Identify how the program will retrieve data from the database and update it. Make decisions about the following:

- Primary or secondary key(s) to be used
- Index files to be used
- Program input and output formats
  - Loading data into tables or arrays
  - Use of leading dollar signs
  - Decimal accuracy

Develop Structure Charts

Document the hierarchical arrangement of the program in the form of program structure chart(s). Provide the name and number of each module, and indicate the modules that are called by each module higher in the structure of the program. Use project naming standards and configuration item identifiers.

Cross-Reference with Program Specifications

Provide a means by which the program specifications can be cross-referenced to the structure charts to provide program traceability.

Inspect Structure Charts

After the structure chart is developed and the modules are cross-referenced with the program specifications, perform a QA review of the structure chart to ensure that all logic documented in the program specifications is accounted for in the structure chart of the program under development. Modify the structure chart, as necessary, to resolve any discrepancies found during the review.

4.3.2 Develop Programs

Develop the programs in the programming language(s) selected for the project. Use the database specifications and the program specifications when developing each program. If a CASE tool is being used, use the tool to generate code from the specifications. Follow coding standards in accordance with HUD and project standards.

Code Programs

Write source code instructions in the selected programming language that will perform the logic documented in the program specification. For each program, follow the input and output layouts depicted in the data access portion of the program specification and follow the structure chart while developing the program. Use coding standards appropriate for the programming language. Add comments into the code at appropriate locations to provide clarification of the function and logic being performed. The types of software units and programs include the following:

- ***Utility Programs.*** Programs that are not available as part of the programming language's compiler or are not commercially available but that contain common logic that will be used by other programs in the system. Utility programs may include edit or sort routines, control of user access to functions, or error message handling routines.
- ***Input and Output Modules*****.** Programs that handle the retrieval or output of data for the system. These programs may handle anticipatory retrieval and caching of data from the database for output, deferred storage, or data formatting.
- ***Application Programs.*** Programs that perform the actual logic or "mainline" of the system being developed.
- ***Command Language*****.** Programs that are written in operating system command language or job control language and that are used to execute batch jobs, compile programs, execute programs, and control peripheral devices.

Compile Programs and Correct Errors

Use the control programs developed along with the source programs to submit the source code for compilation. Compile the program source code with error and syntax checking.

Using the source listing report, identify any error messages. Modify the source code to correct the problem. Recompile the program after the changes have been made, and recheck the source listing. Repeat the process until no error messages are generated.

Perform Code Desk Check

Check the source listing of the program, using a list, checklist, or guide of common errors not typically found by the compiler.

Inspect Code

Perform an inspection of code either one on one between a programmer and an inspector or in a team environment using a moderator and a team as inspectors. The moderator records all problems identified at the inspection and tracks the problems to successful resolution. Perform a one-on-one inspection on programs that are simple and noncritical to the system. Perform a team inspection on programs that are high risk, have security and safety implications, or are central to the successful implementation of the system under development. For team inspections, also inspect, optionally, the program specifications and structure charts with the program source listing. All programs should undergo some type of code inspection.

4.3.3 Unit-Test Programs

After a module has been coded and compiled without error and has undergone an inspection, it is ready to be tested as a standalone entity. This unit testing of software modules and programs uses both valid and invalid data developed specifically for the execution of the test. Unit testing employs dynamic testing techniques that execute the software module with both expected and erroneous data and compares actual with expected results. The goal of unit testing is to exercise all functions of the software module and all logic paths within the module.

Develop Unit Test Procedures

Document a procedure for each unit test to be executed for the program. Each unit test procedure should contain the following information:

- Name of the software module to be tested
- Description and objective of the test
- Any test stubs or drivers to be used in executing the test
- Test data to be used in the test
- Job control language to be executed
- Expected results
- Steps to be taken to execute the test

Document the unit test plan(s) in accordance with HUD SDM documentation standards (Handbook 2400.15) for Test Plans.

Create Unit Test Data

Employ utilities provided by the DBMS, Structured Query Language (SQL), or other data manipulation tools to develop the data necessary to execute the unit tests by either adding new data to the database or modifying data that already exist in the database.

Execute Unit Tests

Once the test procedures and test data have been prepared, the software unit undergoes the test cycle shown below:

- ***Run Test.*** Follow the steps necessary to execute the test as documented in the unit test procedure. If any additional steps are found to be necessary during the test execution, perform the steps and update the test documentation to reflect the modification(s).

- ***Verify Results and Correct Errors.*** After the test has been executed, compare the actual tests results with the expected results documented in the unit test plan. For any unacceptable test results that require program modification, make the modification(s) to the program.
- ***Rerun Test.*** Repeat the test cycle until the program successfully passes all tests at the unit level, then promote the program for integration testing.
- ***Inspect Unit Test.*** Perform a quality review of the results of unit testing to ensure that all tests were successfully conducted, all logic paths were exercised, and any known problems or errors were adequately documented.

4.4 INTEGRATE DATABASE AND PROGRAMS

Develop tests to ensure that all software unit and program interfaces are present and function correctly. After a program has successfully completed unit testing, add it to a group by interface and test those programs in the group together. As groups of programs test successfully, integrate them with other groups and test all as a whole. Continue grouping and testing until the entire system has been integrated and tested. If any program fails, rework and unit test it; and then repeat each prior integration test. If integration testing requires access to the production environment, coordinate with Operations to minimize the impact and reserve test time. Include the following activities:

4.4.1 Define integration test.

4.4.2 Create test data.

4.4.3 Execute integration tests.

4.4.4 Document all test results.

Table 4-6 defines the roles and responsibilities of key personnel for conducting integration testing.

Table 4-6. Roles and Responsibilities for Conducting Integration Testing

Role	Responsibility
Project Manager	Monitors integration testing.
Developer	Prepares integration test plans. Integrates and tests computer programs and database. Updates requirements matrix.
Configuration Management	Establishes controlled environment for integration test.
Quality Assurance	Monitors activities for conformance to HUD and project standards and guidelines.

4.4.1 Define Integration Test

The Integration Test Plan should provide the coordination necessary for developers to define, document, and execute each integration test identified and to develop the test data required to perform the test. The Integration Test Plan should identify the following:

- Each integration test to be executed
- System and program requirements to be tested
- Location of each test in the hierarchy of the testing to be performed
- Software units and programs that are to be included in the test
- Required test data, including database fields
- External (or existing) programs needed to support the test
- Procedures for reporting errors, test results, and reworking and retesting programs

Document the Integration Test Plan in accordance with HUD SDM documentation standards (Handbook 2400.15) for Test Plans. Perform a quality review of the Integration Test Plan. Ensure the plan identifies the program functions and interfaces that require testing.

4.4.2 Create Test Data

Identify required test data, either converted or input from an existing system. Use utilities provided by the DBMS, SQL, or other data manipulation tools to modify test data, as necessary, to create erroneous data as well as "staged" data to test all program interfaces. In the absence of any converted data, use data manipulation tools to create the test data or "test bed." Coordinate with the appropriate operations organization to allow for the recovery of the original data via planned backups of the test bed before test execution. Use CM version control procedures to identify and maintain test data.

4.4.3 Execute Integration Tests

Once the test procedures are developed as identified in the Integration Test Plan and the test data have been prepared, the program undergoes the test cycle listed below:

- ***Run Test.*** Follow the steps necessary to execute each test as documented in the Integration Test Plan. If any additional steps are found necessary during test execution, perform the steps and update the test documentation to reflect the modification(s).
- ***Verify Results and Correct Errors.*** After the test has been executed, perform a review to compare the actual test results. Ensure that actual results conform to the expected results documented in the Integration Test Plan. For any unacceptable test results that require program modification, make the necessary modification(s) to the program(s).
- ***Rerun Test.*** Repeat the test cycle until the program successfully passes all tests at each level in the integration of the system. Once all of the programs in the system successfully pass all integration level tests, promote the system for system acceptance testing.

4.4.4 Document All Test Results

Project test personnel track the results of all tests that are identified in the Integration Test Plan and are executed during testing. At the completion of testing, record and summarize test results in a Test Analysis Report. Along with the test results, update the requirements matrix to indicate that a test has verified each functional requirement. Add the matrix to the Test Analysis Report. Provide a summary of the system's capabilities, identified deficiencies, and urgency for correcting each deficiency. Integration test personnel provide to project management an assessment of the system's readiness for continuing to the next phase.

4.5 DEVELOP OPERATIONS AND MAINTENANCE DOCUMENTATION

Develop, review, and approve all procedural documentation necessary for installation, conversion, operation, maintenance, and use of the system. Update existing office procedures or create new ones to accommodate any impact that may occur from operation of the developed system. Include the following activities:

4.5.1 Develop user procedures.

4.5.2 Develop operations procedures.

4.5.3 Develop office procedures.

4.5.4 Develop maintenance procedures

4.5.5 Develop preliminary installation and conversion procedures.

Table 4-7 defines the roles and responsibilities of key personnel for developing operations and maintenance documentation.

Table 4-7. Roles and Responsibilities for Developing Operations and Maintenance Documentation

Role	Responsibility
Project Manager	Monitors preparation of documentation.
User	Assists in preparing the User's, Maintenance, and Operations manuals and Installation and Conversion Plan.
Developer	Prepares the User's, Maintenance, and Operations manuals and Installation and Conversion Plan.
Computer Services	Reviews the User's, Maintenance, and Operations manuals and Installation and Conversion Plan.
Quality Assurance	Reviews documentation for adherence to HUD and project standards and guidelines.

4.5.1 Develop User Procedures

Develop a User's Manual that provides system procedures necessary to allow user organizations to determine the software's applicability and when and how to use it. Update current office procedure documentation to reflect changes that will occur as a result of the system becoming operational.

Prepare System Overview

Develop an overview of the system detailing the intended purpose of the system and outlining how that purpose is met by the system. Include in this outline a description of the following:

- Intended purpose of system(s)
- Relationships among inputs, outputs, and system functions
- Equipment, communications, and networks

- System performance capabilities
- Contingencies and alternate modes of operation
- Method used to store and maintain data
- Data flows

Describe Input Procedures and Expected Output

Prepare a detailed series of instructions (in nontechnical terms) describing the procedures the user will need to follow to use the system. The following information should be included in these instructions:

- Detailed procedures to initiate system operation, including identification of job request forms or control statements and the input’s frequency, reason, origin, and medium for each type of output
- Illustrations of input formats
- Descriptions of input preparation rules
- Descriptions of output procedures identifying output formats and specifying the output's purpose, frequency, options, media, and location
- Identification of all codes and abbreviations used in the system's output
- Description of all recovery and error correction procedures, including error conditions that may be generated, corrective actions that may need to be taken, and any recovery and restart procedures

Describe File Query Procedures

Develop detailed descriptions of the procedures necessary for file query including the parameters of the query and the sequenced control instructions to extract query requests from the database.

Update Office Procedures

Update or create office procedure documentation to reflect new procedures to be instituted upon release of the system into operation.

Prepare User's Manual

Document user procedures in the system User’s Manual, including the system description, input and expected output procedures, user interface descriptions, file query, and office procedures. Describe this information to allow user organizations to determine the system’s applicability and when and how to use it.

Prepare the User’s Manual in accordance with HUD SDM documentation standards (Handbook 2400.15).

4.5.2 Develop Operations Procedures

Develop an Operations Manual to provide computer operations personnel with a description of the software, the environment in which this software will be run, and the procedures to be followed.

Return to Title Page

Prepare System Overview

Develop an overview of the system that details its intended purpose and outlines how that purpose is met by the system. Include the following in the outline:

- Description of the system operation
- Inventory of all software units
- List of all permanent files and databases referenced
- List of all reports
- Process overview that includes descriptions of system restrictions, any waivers of operational standards, and all interfaces with other systems
- Description of the communications network within the system
- Description of security considerations associated with the system

Describe System Runs

Develop descriptions for all system runs with accompanying run listings and operation schedules, and define the setup and diagnostic procedures for any software diagnostics. Define software units and jobs by run, with a list of all error messages and any corresponding correction procedures. Provide the following information in the run descriptions:

- Purpose of each run
- Runstream job control statements for job initiation
- Run management requirements
- Descriptions of all related files and databases
- Requirements and procedures for report generation and reproduction
- Any restart and recovery procedures

Prepare Operations Manual

Document all operations procedures in the system's Operations Manual, including the system description and system run procedures. Ensure that the manual is well indexed and cross-referenced and includes a glossary of jargon, special terms, and acronyms.

Prepare the Operations Manual in accordance with HUD SDM documentation standards (Handbook 2400.15).

4.5.3 Develop Office Procedures

Update or create Desk Procedures Manuals to provide detailed descriptions of the changes to office procedures made necessary by the introduction of the system. Include a full description of requirements related to security, privacy, and internal controls.

Review and Update Office Procedures

Review current office procedures to determine the impact that operation of the new system will have on the office environment. Consider the following when performing this review:

- Manual procedures that have been automated as a result of the new system
- Procedures that have become obsolete
- New procedures, both manual and automated, that are now required because of the operational release of the new system

Based on the results of the review, update or develop office procedures to meet the requirements of the new system.

Create and Update Desk Procedures Manuals

Document all information in desk procedures manuals concerning changed, updated, or newly created office procedures and their related interface with the system. Ensure that each manual is well indexed and cross-referenced and includes a glossary of jargon, special terms, and acronyms.

Prepare all documentation in accordance with HUD documentation standards.

4.5.4 Develop Maintenance Procedures

Develop a Maintenance Manual to provide the maintenance programmer the information and source code necessary to understand the programs, operating environment, maintenance procedures, and security and control requirements.

Prepare System Overview

Develop an overview of the system that details the purpose of the system and outlines how that purpose is met by the design of the system. Include the following in this outline:

- Explanation of the system’s purpose and its functions
- Description of system, subsystems, and communications design
- Discussion of security considerations associated with the system

Describe System Environment

Prepare a detailed description of the environment in which the system is to be maintained. Include the following information in this description:

- Configuration of equipment
- Listing of support software identified by version and release number
- Purpose and content of system database
- Database characteristics, such as IDs and storage
- Identification of data structures, elements, and entities

Describe Software and Database Maintenance Procedures

Prepare a detailed description of procedures to be followed by the maintenance programmer. Include the following information in this description:

- Identification of all system software units

- Description of each software unit that includes:
  - Description of each functions
  - Illustration of each input format
  - Detailed descriptions of processing
  - Description of each output produced
  - Amount and type of storage required
  - Interfaces
- Description of the database rules, schema, and conventions used
- Description of performance verification procedures
- Description of all error conditions and correction procedures
- References to location of software listings

Prepare Maintenance Manual

Document in a Maintenance Manual all information concerning the maintenance programmer's interface with the system, including system overview and software maintenance procedures. Ensure that the manual is well indexed and cross-referenced and includes a glossary of jargon, special terms, and acronyms.

Prepare the Maintenance Manual in accordance with HUD SDM documentation standards (Handbook 2400.15).

4.5.5 Develop Preliminary Installation and Conversion Procedures

Produce the preliminary installation and conversion procedures to direct installation or implementation of an automated information system at locations other than the test site after testing has been completed.

Determine System Installation Procedures

Develop procedures that must be followed when the system is installed. Include an overview of the installation procedure, identifying the following information:

- Support materials needed
- Personnel requirements
- Security requirements
- Special training needs, if any

Provide in the overview additional details for pilot site and operations site installations. Include the following information in those details:

- Installation schedule
- Inventory of hardware and software to be installed
- Operation facility requirements

- Identification of installation team
- Procedures to be followed during installation

Determine Data Conversion Procedures

Develop conversion procedures describing the initiation procedure and output formats. Identify sequential steps to perform the conversion and describe all recovery and correction procedures.

Summarize Preliminary Installation and Conversion Strategy

Prepare a preliminary Installation and Conversion Plan describing the system installation procedures and the data conversion procedures. Present in nontechnical terms those portions of the document directed toward staff personnel; present in suitable terminology those parts written for operations personnel.

Prepare the Installation and Conversion Plan in accordance with HUD SDM documentation standards (Handbook 2400.15).

4.6 FINALIZE TEST STRATEGY

Combine the initial test strategy developed during the Design System phase with all system and testing information developed during the Build System phase to prepare the final VV&T Plan. The plan is reviewed and approved by the appropriate areas within HUD. Include the following activities:

4.6.1 Refine test strategy.

4.6.2 Finalize VV&T Plan.

Table 4-8 defines the roles and responsibilities of key personnel for finalizing the testing strategy.

Table 4-8. Roles and Responsibilities for Finalizing Test Strategy

Role	Responsibility
Project Manager	Monitors test activities.
User	Reviews test strategy and test plans.
System Acceptance Test Team	Finalizes test strategy. Finalizes VV&T Plan.
Computer Services	Provides input to the system test strategy.
Configuration Management	Maintains changes to test data.
Quality Assurance	Reviews VV&T Plan for adherence to applicable standards.

4.6.1 Refine Test Strategy

Update the test strategy to include any revisions resulting from Build System phase activities. Ensure that the test database is updated to meet any changes in test requirements or procedures or both.

Return to Title Page

Revise Test Strategy

Update the description of the test strategy to reflect changes that may have occurred during the Build System phase.

Revise Test Data

Update the test database to reflect changes to the test strategy. Make sure that sufficient data have been developed to ensure the performance of all required tests. Ensure version control procedures are used to manage changes incorporated into test data.

Revise Test Procedures

Update the system acceptance test procedures developed during the Design System phase to reflect any impact on the testing strategy that may have occurred during the Build System phase.

4.6.2 Finalize Validation, Verification, and Test Plan

Perform final updates to the VV&T Plan to reflect any changes in system acceptance testing requirements that may have occurred during the Build System phase of development.

Update VV&T Plan

Revise the VV&T Plan to include changes to the testing schedule and equipment or personnel requirements. Verify that the VV&T Plan accurately reflects the materials needed to perform system acceptance testing.

Update Test Specifications

Revise the VV&T Plan to include any changes to the test specifications. Review the following to ensure that they accurately reflect the current specifications:

- Performance requirements
- Test methods
- Test conditions
- Test data criteria

Prepare the VV&T Plan in accordance with HUD SDM documentation standards (Handbook 2400.15).

4.7 COMPLETE TRAINING DOCUMENTATION

Finalize the strategy for training the user community and all related system personnel (e.g., operators and support personnel). If a training guide is to be used to assist instructors in their training tasks, finalize the training guide as well. Include the following activities:

4.7.1 Finalize training approach.

4.7.2 Finalize training guide (optional).

Table 4-9 defines the roles and responsibilities of key personnel for completing training documentation.

Table 4-9. Roles and Responsibilities for Completing Training Documentation

Role	Responsibility
Project Manager	Monitors training activities.
User	Finalizes Training Plan.
Developer	Prepares Training Plan. Develops training guides, if necessary.
Quality Assurance	Monitors training activities for adherence to HUD and project standards.

4.7.1 Finalize Training Approach

Update the approach to the training activities to include any changes necessitated by new information that was received during the Build System phase of development.

Identify Modifications

Prepare a list of the changes that need to be made to the training approach to reflect the impact of any changes that may have occurred during the Build System phase of development. Consider the following when identifying modifications that must be made:

- Training schedules
- Student attendance lists
- Training requirements for office procedures
- Training materials

Document Modifications

Prepare a finalized Training Plan that reflects any modifications made to the training strategy. Ensure that the finalized Training Plan is developed in accordance with HUD SDM documentation standards (Handbook 2400.15).

4.7.2 Develop Training Guide (Optional)

Develop a guide to assist in instructing various personnel (e.g., management, operations, programmers, and users) in the use of the system.

Determine Training Objectives

Based on a review of the roles of the personnel to be instructed, determine the objectives that must be met for the students to achieve an adequate understanding of the system functions to perform their job functions with minimum guidance. Objectives may vary depending on the skill of the class participants and their levels of involvement with the system.

Prepare Training Guide

Outline the procedures for instructing various types of personnel in the use of the system. Develop a training guide for each personnel group to include lesson plans, objectives to be met, equipment to be used (e.g., overhead projectors and classrooms), procedures to be followed

Return to Title Page

(e.g., Computer-Based Training (CBT)), and a course book for the students to follow during their training modules, if necessary.

4.8 UPDATE SYSTEM AUDIT STRATEGY

Notify the Inspector General (IG) staff about any work products completed during the Build System phase. Inform them about any changes that may have an impact on internal data processing controls as described in the Build System phase documentation. Include the following activities:

- ***Coordinate with IG Staff*****.** Schedule meeting(s) with the IG staff to inform them of any developments that occurred during the Build System phase. Determine the scope of the information required from the project manager for the IG staff to perform the necessary updates to the Internal Audit Plan.
- ***Provide input to IG Staff*****.** Provide the IG staff the necessary information.
- ***Update Internal Audit Plan*****.** The IG staff updates the Internal Audit Plan to include information provided by the project manager.

Table 4-10 defines the roles and responsibilities of key personnel for updating system audit strategy.

Table 4-10. Roles and Responsibilities for Updating System Audit Strategy

Role	Responsibility
Project Manager	Provides project information to the IG staff, as requested.
IG Staff	Coordinates with project manager, as necessary. Updates Internal Audit Plan.

4.9 UPDATE SYSTEM DECISION PAPER

Update the System Decision Paper to include a summary of the progress made during the Build System phase and the schedule of events for Build System phase activities. Include the current status of the project. List in detail any changes to the original plan for development of the project and the parties responsible for approving those changes. The updated System Decision Paper is then reviewed and approved by the appropriate level of management. Activities include the following:

4.9.1 Summarize progress of system.

4.9.2 Identify changes.

4.9.3 Summarize schedule of events.

4.9.4 Summarize status.

4.9.5 Document results.

Table 4-11 defines the roles and responsibilities of key personnel for updating the System Decision Paper.

Return to Title Page

Table 4-11. Roles and Responsibilities for Updating System Decision Paper

Role	Responsibility
Project Manager	Updates System Decision Paper, as necessary.
Developer	Provides inputs needed to update System Decision Paper.
Quality Assurance	Reviews System Decision Paper for adherence to standards.

4.9.1 Summarize Progress of System

Summarize the progress of project development through the Build System phase and update the System Decision Paper accordingly.

4.9.2 Identify Changes

Document any changes to the system development milestones and schedule, and identify any modifications made to the project strategy. Update the System Decision Paper accordingly.

Describe Changes in Milestones, Schedules, and Tasks

Provide a description of any Build System phase occurrences that may have an impact on projected milestones, schedules, and tasks of the development effort. Provide a brief explanation of the nature of these impacts, and describe any changes to the milestones or schedule that may result from these impacts.

Describe Modifications to Project Strategy

Provide a description of any modifications to the project strategy that have occurred during the Build System phase of development. Provide a brief explanation of these modifications, indicating the reasons for the changes and any anticipated impacts.

4.9.3 Summarize Schedule of Events

Develop a brief summary of events that took place during this stage of development and a description of the schedule of development activities that remain. Show the impact these events have had on the original schedule and any anticipated impacts on future development activities.

4.9.4 Summarize Status

Provide a status summary on the system’s development, including what percentage of the system has been developed, has successfully passed integration testing, and is ready for acceptance testing, as reported by the integration test team in the Test Analysis Report. Indicate the components or subsystems that may be causing delays or problems.

4.9.5 Document Results

Document all information for the phase in the System Decision Paper in accordance with HUD SDM documentation standards (Handbook 2400.15)

Return to Title Page

4.10 REFINE PROJECT PLAN

The Project Manager controls the project by monitoring phase activities, taking corrective action where necessary, and refining the project plan to account for changes attributable to actions taken in the current phase and new information for upcoming phases. The Project Manager also reviews the quality process planned by users, developers, testers, and quality assurance for the next phase. Activities include the following:

4.10.1 Update Project Plan.

4.10.2 Review planned quality process.

Table 4-12 defines the roles and responsibilities of key personnel for refining the Project Plan.

Table 4-12. Roles and Responsibilities for Refining the Project Plan

Role	Responsibility
Project Sponsor	Approves changes to the project schedule. Approves changes to the Project Plan.
Project Manager	Updates the Project Plan to include actual costs incurred and specific activities accomplished for the Build System phase; and revises project schedule, plans, strategies, resources and requirements for subsequent phases, as required. Reviews, for approval, the quality process planned by users and developers for the next phase.
User	Provides updated input to project manager on required Evaluate System phase activities. Works with developer to review and update the quality process planned for the next phase. Attends the Project Plan review to provide technical input to the review process, as required.
Developer	Provides updated input to project manager on required Evaluate System phase activities. Works with user to review and update the quality process planned for the next phase.
Quality Assurance	Provides updated input to the Project Plan.
Configuration Management	Provides updated input to the Project Plan.

4.10.1 Update Project Plan

Update the Project Plan with cost, schedule, and budget data for the current phase (Build System) to the level of detail necessary to reflect the status of the project. Review the plan and determine whether any activities described for the next phase (Evaluate System) and subsequent ones are affected by the completion status of the current phase activities. Adjust the schedules and resource requirements for activities in the next phase, if necessary, and assign starting and ending dates for activities that have been affected. Take into account that the starting date for some activities may be dependent on the completion of other activities. Update milestones, schedules, and resource requirements for the remainder of the project.

4.10.2 Review Planned Quality Process

Users, developers, testers, and QA work closely together to determine the process to be used to build the product with the desired quality. Activities include tailoring the planned quality process for the next phase and identifying the standards and procedures to be used. The project manager reviews the quality process for approval and execution in the next phase.

4.11 REVIEW AND APPROVE DOCUMENTS AND DELIVERABLES

The review and approval of documents and deliverables is an iterative process as each phase provides more definition and products are revised. Additionally, the approval process offers senior management the capability to monitor the activities of the project. Activities include the following:

4.11.1 Review revised products from prior phases, as required.

4.11.2 Conduct review for Build System phase documents and deliverables.

4.11.3 Obtain approval of project documents and deliverables.

Table 4-13 defines the roles and responsibilities of key personnel for updating the System Decision Paper.

Table 4-13. Roles and Responsibilities for Review and Approval of Deliverables and Documents

Role	Responsibility
Project Manager	Determines required review level and schedules reviews. Determines needed changes to products of prior lifecycle phase. Attends reviews and presents deliverables. Obtains concurrence and approvals for deliverables and associated management summaries.
User	Assists in preparation of deliverable reviews. Assists in revising products of prior lifecycle phase. Attends deliverable reviews.
Developer	Prepares deliverable documents. Provides technical expertise during review process. Assists in revising products of prior lifecycle phase.
ADP Security	Reviews Build System phase deliverables to ensure security requirements have been incorporated. Reviews revised products to ensure security standards and guidelines have been addressed.
Project Sponsor	Participates in deliverable reviews (optional).
Computer Services	Participates in review of documents and deliverables.
Configuration Management	Assists project manager with review preparation.
Quality Assurance	Ensures all review procedures are followed, as required. Reviews Build System phase deliverables to ensure they meet all applicable HUD and project standards.

4.11.1 Review Revised Documents from Prior Phases as Required

Review the activities performed during the Build System phase to determine whether they have an impact on any documents produced during the previous phases. If changes are required to documents of a previous lifecycle phase, update these documents to reflect current project developments. Project personnel review revised document in a manner similar to reviews for the current phase documents to ensure that changes are within the scope of the project’s requirements and are in compliance with HUD and project standards and procedures. Prepare a management summary for each revised document that summarizes the essential revisions. Submit the revised products to the appropriate review board for approval. Use the appropriate document review checklists in Appendix E to assist in the reviews. File affected product change records with CM for appropriate version control updates.

4.11.2 Conduct Review for Build System Phase Documents and Deliverables

Conduct a project review with project personnel and system stakeholders to ensure that the project documents and deliverables for the Build System phase include the necessary level of detail, fulfill the system's requirements, and meet the appropriate HUD and project standards and guidelines. Use the appropriate document review checklist in Appendix E for Build System phase deliverables to aid in the document reviews.

A minimum of 10 working days before the scheduled review, notify the personnel required to attend and provide each with a copy of the product for their pre-review. Discuss all comments or objections that are raised during the review, and reach a consensus on one of the following before the review session terminates:

- The document is correct and complete, as is, without any further changes.
- Additional changes that need to be made are minor and do not require further review. In this case, the updates should be made and change pages should be distributed by an agreed-upon date.
- Required changes will have a major impact on the plan. A second review must be scheduled. The changes must be incorporated and the resulting change pages distributed at least 10 working days before the second review.

Prepare a management summary for each document that includes the essential data collected in the document, lists conclusions that may be drawn from the document, and describes the potential impacts on the project, if applicable. Submit the management summaries to the appropriate review board for approval. Documents for approval include the following:

- Database Specifications
- User's Manual
- Operations Manual
- Maintenance Manual
- Desk procedures manual
- Installation and Conversion Plan
- Training Plan
- Validation, Verification, and Test Plan
- System Decision Paper (updated)
- Project Plan (updated)

4.11.3 Obtain Approval of Project Documents and Deliverables

Present the project documents and deliverables to the chairperson of the appropriate review board for approval at least 10 days before the scheduled decision date. Include the management summary information, approval (sign-off) record, System Decision Paper, and any other required or requested information. The review board chairperson coordinates review board comments, recommendations, and approval signature(s) and returns the approval record to the project manager. Recommendations on the approval record are addressed by the project and the document should be resubmitted to the board if requested. Approval will be assumed if there is no response, or the response is “no comment.” Project approval records are maintained by the project’s configuration management (CM) function. A copy is inserted into the central project library.

The project proceeds to the next phase after all project documents and deliverables are approved.

4.12 PERFORM CHANGE CONTROL ACTIVITIES

Any products developed during the Build System phase are baselined and subjected to version control. Products baselined during prior activities are assigned new version control numbers when they undergo change (e.g., an update or rewrite). CM reports are provided to the project manager as requested or required.

Table 4-14 defines the roles and responsibilities of key personnel for performing configuration change control activities for approved baselined products.

Table 4-14. Roles and Responsibilities for Performing Change Control Activities

Role	Responsibility
Project Manager	Approves controlled products for distribution.
Configuration Management	Performs change control for new and revised products. Provides status reports to project management as requested or required.
Quality Assurance	Audits products to ensure only approved changes are addressed.

The CM function prepares the baseline for configuration control. The baseline includes any new technical and document deliverables that will comprise the project configuration baseline. For the baselined items, CM performs the following activities:

- ***Verify changes made to product***. Review the updated product to ensure that the changes have been made as described in supporting documentation. Supporting documentation may be comments received from document reviews. Supporting documentation for software may be software change requests or discrepancy reports generated during testing.
- ***Assign version number***. The version number must follow conventions established by HUD and enable the project's CM to monitor updates to the product and assist in its distribution. For software programs, increment the version number each time a change is made to the software to correct deficiencies found during testing.
- ***Store approved version in central library***. After each baselined product has been completed and approved, according to HUD procedures, store the approved version in the project’s central library. The project's CM function controls access to the library. For software, the library can be a subdirectory or dataset where all baselined software configuration items will be stored.

- ***Record product information in inventory log***. Maintain an inventory log that includes the title of the product, release date, version number, name and version or model numbers of the software and hardware used in the development of the product, name of the organization responsible for development of the product (usually the sponsoring organization), and the product distribution list.
- ***Distribute copies of products as required***. The project's CM function distributes copies of the products according to a distribution list maintained as part of the inventory log information, with distribution based on need and the security level of the product.
- ***Archive old versions of products***. Archive and retain outdated versions of all products for the required period of time, in keeping with HUD standards and guidelines.

Return to Title Page

Section 5. Evaluate System

5.0 Evaluate System

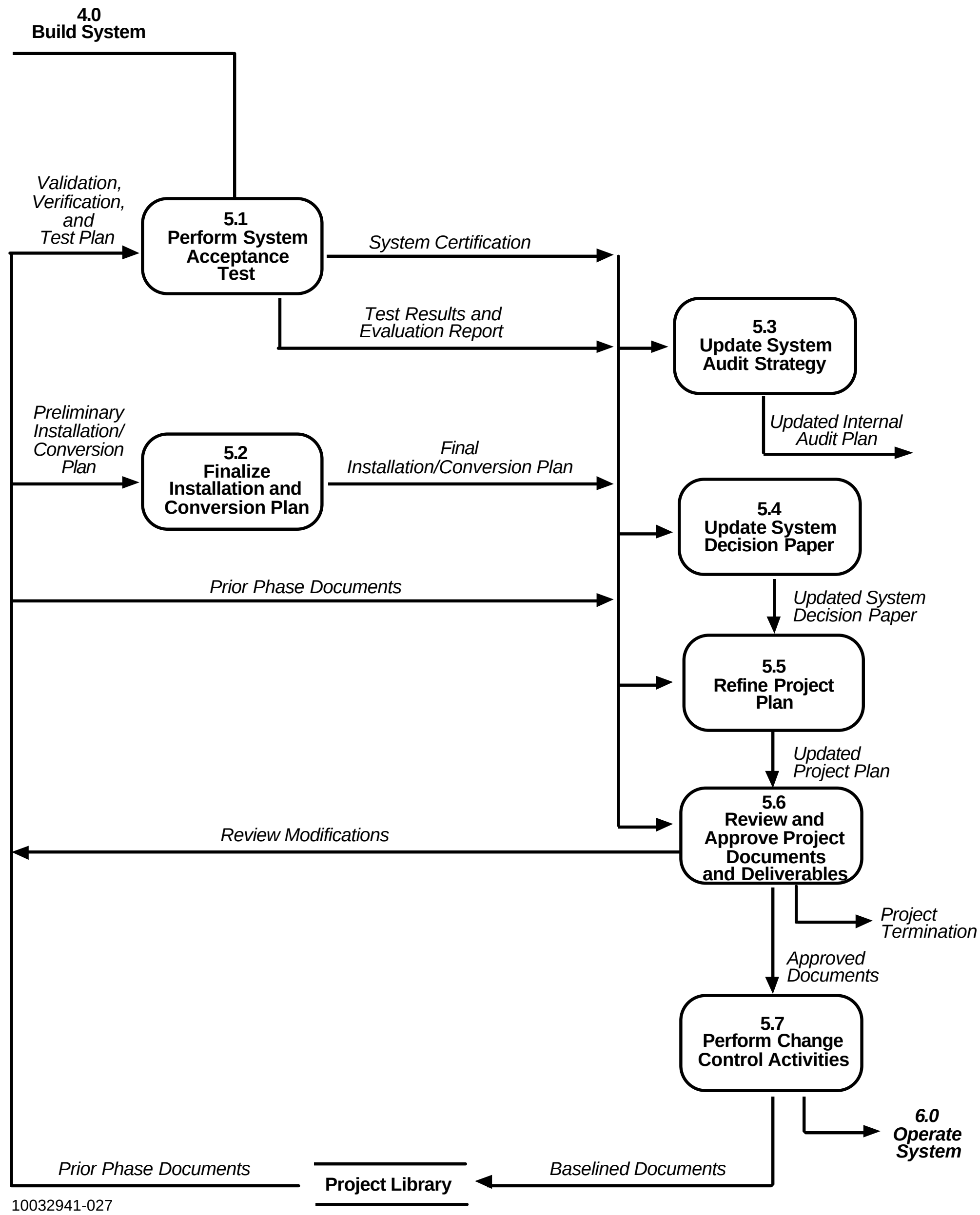

Figure 5-1. Process Flow for Evaluate System Phase

5.0 EVALUATE SYSTEM

Purpose

The Evaluate System phase is the period in which independent testers measure the system's ability to perform the functions that are required by the user and ensure an acceptable level of performance. After this phase of development is completed, a clear indication of the system's readiness for operation is evident. Figure 5-1 highlights the process flow for the Evaluate System phase.

Overview

During the Evaluate System phase, the system undergoes a complete and thorough System Acceptance Test conducted by a test team that is designated by the project sponsor and is independent of the software development organization. Each of the system acceptance test procedures and scenarios, documented in the VV&T Plan, is executed. The results of the system acceptance test, as well as the testing methods employed, are documented for final approval.

During the Evaluate System phase, conduct the following activities:

- Execute each test according to steps documented for the test's procedure or scenario in the VV&T Plan.
- Record and track results of the tests, and compare them against expected results.
- Document and report any errors detected using the project's error reporting and correction procedures. Retest corrections submitted by the project development team.
- Once all tests are executed, compile and document the results, along with a summary of the system's readiness for production, in the Test Results and Evaluation Report.
- Submit the report for review by appropriate project personnel and review boards.
- Perform functional and physical configuration audits.
- Finalize the Installation and Conversion Plan based on any modifications necessary to the system as a result of testing; address all applicable installation and conversion procedures, including pilot and production sites.
- Revise, as necessary, all deliverables produced during prior phases of development to reflect changes required by the system as a result of testing or to reflect changes in strategy as a result of activities performed during testing.
- Track all deliverable products produced using the project's CM procedures, to include placing products under configuration control when originally produced and controlling changes when modifications to the controlled items are made.
- Perform project reviews of all products shown in Table 5-1; obtain approval of Evaluate System phase products from the appropriate review boards.

- Present the System Decision Paper and supporting documentation to the appropriate review board at the end of this phase. Projects that receive review board approval continue to the next phase, the Operate System phase.

Table 5-1. Evaluate System Phase Functions and Products

Evaluate System Functions	Products
5.1 Perform Systems Acceptance Test	Test Results and Evaluation Report
5.2 Finalize Installation and Conversion Plan	Installation and Conversion Plan (final)
5.3 Update System Audit Strategy	Internal Audit Plan (updated)
5.4 Update System Decision Paper	System Decision Paper (updated)
5.5 Refine Project Plan	Project Plan (updated)
5.6 Review and Approve Documents and Deliverables	New and revised products Management Summary[1]
5.7 Perform Change Control Activities	Change control records

[1] A management summary is prepared for each product produced or revised during the Evaluate System phase. This one-page summary includes a summary of the essential data collected in a document product, conclusions that may be drawn from the document, and potential impacts on the project, if applicable.

Standards and Guidelines

Follow HUD SDM documentation standards (Handbook 2400.15) and project guidelines during development of the Evaluate System phase products.

Roles and Responsibilities

Throughout the Evaluate System phase of development, key personnel are required to perform the various tasks and activities outlined in the SDM. Table 5-2 lists types of personnel required and the activities for which they are responsible.

Table 5-2. Roles and Responsibilities for Evaluate System Phase (1 of 3)

Role	Responsibility
Project Sponsor	Provides input and approves changes to project schedule and Project Plan. Provides input to Installation and Conversion Plan. Reviews and approves system certification.
Project Manager	Assists activities of the system acceptance test team. Reviews and approves system certification. Issues Release Request.

Return to Title Page

Table 5-2. Roles and Responsibilities for Evaluate System Phase (2 of 3)

Role	Responsibility
Project Manager (Cont'd)	Coordinates with IG staff, as required. Updates System Decision Paper, as necessary. Determines needed changes to prior phase documents and ensures revision in accordance with the overall Project Plan. Coordinates review of Evaluate System phase deliverables. Determines level of required review and schedules required reviews. Attends reviews and presents deliverables. Updates Evaluate System phase deliverables to include any recommendations received during review. Obtains appropriate concurrences and approvals for Evaluate System phase document deliverables and associated Management Summaries. Obtains appropriate concurrences and approvals of updated documentation. Updates Project Plan to include actual costs incurred and specific activities accomplished for the Evaluate System phase; and revises project schedule, plans, strategies, resources, and requirements for the Operate System phase, as required. Reviews, for approval, the quality process planned by users and developers for the next phase.
Project Development Team (User and Developer)	Determines additional resource and support requirements. Determines additional support requirements. Finalizes Installation and Conversion Plan. Supports System Acceptance Testing. Resolves issues raised during System Acceptance Testing, as required. Reviews Test Results and Evaluation Report and concurs with system certification. Provides input to Project Manager on required Evaluate System phase tasks and activities. Assists in the revision of prior phase products. Assists with preparation of Evaluate System phase deliverables for review. Attends deliverable reviews. Provides technical expertise during the review process. Reviews and updates the quality process planned for the next phase.
Computer Services	Confirms and approves additional project support requirements. Participates in review of Project Plan (optional). Participates in System Acceptance Testing.

Table 5-2. Roles and Responsibilities for Evaluate System Phase (3 of 3)

Role	Responsibility
Computer Services (Cont'd)	Reviews Test Results and Evaluation Report and concurs with system certification. Revises capacity planning estimates, as required. Approves Release Request.
ADP Security	Reviews Evaluate System phase deliverables to ensure that all necessary security requirements have been thoroughly tested. Reviews revised products to ensure that the necessary security standards and guidelines have been addressed. Reviews Test Results and Evaluation Report to ensure that all necessary security requirements have been addressed.
IG Staff	Coordinates with project manager as necessary. Updates internal audit plan to incorporate information received from the project manager.
Quality Assurance	Reviews revised products to ensure they meet all applicable HUD standards and guidelines. Ensures that all review procedures are followed, as required. Reviews Evaluate System phase deliverables to ensure they meet all applicable HUD standards and project guidelines. Coaches users and developers in the planning for quality in the next phase.
Configuration Management	Establishes change control over new hardware and software configuration items. Performs version control for revised hardware and software. Provides CM reports to project management as requested or required. Assists the project manager in preparing Evaluate System phase deliverables for review.
System Acceptance Test Team	Updates system acceptance testing strategy. Performs system acceptance testing. Prepares system certification. Provides technical expertise during review process. Develops Test Results and Evaluation Report. Updates requirements matrix.
Project Data Administrator	Documents final logical database.
Project Database Administrator	Participates in review of Project Plan (optional). Provides control over test data and test database. Provides monitoring and tuning of applications.

5.1 PERFORM SYSTEM ACCEPTANCE TEST

Until this point, all testing has used procedures and scenarios generated by the project development team and has emphasized correct operation as specified in the analysis and design documentation. System acceptance testing emphasizes the proper functioning of the system from the user's point of view. System acceptance testing is the process of demonstrating whether the program meets the user's written set of measurable objectives. The primary focus of the test should be on translation errors, which are errors made in the process of transforming objectives and requirements of the system (documented in the Functional Requirements Document and the Data Requirements Document) into design specifications (System/Subsystem, Program and Database) and, finally, into an operational system. The classes of tests listed below provide a general description of the testing that is executed during this phase of testing.

- ***Requirements Validation.*** This class of testing ensures that all functional requirements are implemented as originally envisioned by the users and that all requirements are accounted for in the system.

- ***Functional Testing.*** Functional testing stresses the correct operation of the application; the correct addition, modification, and deletion of data from the system; and the correct operation of any audit trails in the system.

- ***Performance/Volume/Stress Testing.*** Performance testing shows whether the program satisfies its performance or efficiency requirements. Volume testing is intended to demonstrate the system's capability to handle the volume of data specified in the requirement. Stress testing determines the maximum capacity of the system, given user requirements for response time and throughput.

- ***Security Testing.*** Security testing checks the adequacy of security processes and procedures by trying to violate those countermeasures. Both authorized and unauthorized transactions and processes are attempted.

- ***Ease of Use.*** Testing the system for ease of use ensures that the system is user friendly, the processes and messages are easy to understand, online documentation is available and useful, the use of function keys is standardized throughout the system, and the presentation of the screens and reports are the same as those approved by the user group during the Define System phase.

- ***Operational Testing.*** Operational testing tests the procedures for installing and operating the systems software on its related hardware. It also tests the backup and recovery procedures.

- ***Documentation Testing.*** Documentation testing evaluates documentation for content, clarity, and consistency. Content refers to the relevance and completeness of the documentation and its applicability to the computer system. Consistency refers to the maintenance of standards throughout the documentation, uniform terminology, and consistency with other documents. The instructions from the user documentation is checked for accuracy against actual operation of the system.

- ***Procedure Testing.*** Procedure testing examines the interface between the programs (system) and any manual systems or human procedures, such as those followed by the system operator, database administrator, or terminal user.

- ***Interface Testing.*** Interface testing evaluates the system's ability to perform required interfaces with other systems operated by HUD and outside organizations.

During this phase of the system's development, final preparations for performing system acceptance testing are undertaken, including finalizing the VV&T Plan and readying the test environment. The requirements matrix is updated to reflect the specific tests used to verify the requirements. Each test identified in the VV&T Plan is executed, and the results of the test are evaluated against expected results. After testing is completed, the Test Results and Evaluation Report is prepared, and the system is certified for production readiness. Reviews are held to determine the quality and thoroughness of the system acceptance testing as documented in the Test Results and Evaluation Report and the requirements matrix. Approval and concurrence for both documents is received from the appropriate review board, and the project sponsor organization certifies the system before its release into production. Activities include the following:

5.1.1 Prepare for system acceptance testing.

5.1.2 Execute tests.

5.1.3 Evaluate results.

5.1.4 Perform configuration audits.

5.1.5 Determine readiness.

Table 5-3 defines roles and responsibilities of key personnel for performing system acceptance testing.

Table 5-3. Roles and Responsibilities for Performing System Acceptance Testing (1 of 2)

Role	Responsibility
Project Sponsor	Designates the independent test team. Reviews and approves system certification.
Project Manager	Assists activities of the system acceptance test team. Conducts physical and functional configuration audits. Reviews and approves system certification. Issues Release Request.
User	Determines additional resource and support requirements. Participates in System Acceptance Testing. Reviews Test Results and Evaluation Report and concurs with system certification.
Developer	Resolves issues raised during System Acceptance Testing, as required. Performs physical and functional configuration audits.
System Acceptance Test Team	Updates system testing strategy. Updates requirements matrix. Performs system acceptance testing. Develops Test Results and Evaluation Report. Prepares system certification.

Return to Title Page

Table 5-3. Roles and Responsibilities for Performing System Acceptance Testing (2 of 2)

Role	Responsibility
Computer Services	Confirms and approves additional project support requirements. Participates in System Acceptance Testing. Reviews Test Results and Evaluation Report and concurs with system certification. Approves Release Request.
ADP Security	Reviews Test Results and Evaluation Report to ensure that all necessary security requirements have been addressed.
Project Data Administrator	Documents final logical database.
Project Database Administrator	Provides control over test data and test database. Provides monitoring and fine-tuning of applications.
Configuration Management	Controls software test environment. Processes changes to software test environment.
Quality Assurance	Monitors conduct of system acceptance testing. Verifies problem reporting procedures. Monitors physical and functional configuration audits.

5.1.1 Prepare for System Acceptance Testing

Undertake final preparations that allow for execution of the VV&T Plan that was finalized and approved during the Build System phase. Make final modifications to the test procedures, scenarios and test data developed during the Build System phase. Ensure that the objectives and requirements are correctly outlined by the project development team in the Functional Requirements Document and Data Requirements Document and the System/Subsystem, Program, and Database Specification documents. Update the requirements matrix to validate that all functional requirements are verified by one or more tests.

Finalize Test Procedures and Scenarios

Modify test procedures and scenarios to reflect any changes to the system that were made to correct errors found during integration testing. Make any necessary changes to the procedures in the following areas:

- Objectives of each test or test description
- Resources needed to execute the test
- Steps to be taken to execute the test
- Expected test results

Prepare Test Data

The test data or testbed that was created for the unit and integration tests performed during the Build System phase may prove sufficient to serve as the system acceptance testing testbed.

Modify the data, if necessary, to allow all test conditions to be executed. Make any necessary final modifications to the test data by using utilities provided by the DBMS, SQL, or other data manipulation tools. Perform the tests as outlined in the test procedure or scenario. Coordinate with the project's Database Administrator (DBA) and with Computer Services as necessary to initialize the database and provide adequate space for the test data. Make any provisions at this time for restoring test data after test executions.

Finalize Test Environment

Before the start of the test, ensure that all necessary resource requirements have been met and are in place at the test site(s). Ensure that the hardware has been installed and that computer time has been made available for the test team (if the tests are to be run using production hardware).

5.1.2 Execute Tests and Verify Results

The standard test cycle includes the activities described in the following sections. Execute each test as stated in each test procedure or scenario, and document the tests in the approved VV&T Plan. After test execution, verify the results of the test against expected results, and formally record the test results. Repeat the test cycle, if needed, for a given program or portion of the system after errors are corrected by the development staff.

Execute Test

For each test, follow the steps necessary to execute each test procedure (or scenario) as documented in the VV&T Plan. If any additional steps are necessary to execute the test, record the additional steps in the test procedure documentation.

Record and Verify Test Results

Upon completion of each individual test identified in the VV&T Plan, compare the actual output generated by the test against the expected output documented in the test procedures and scenarios. If deviations from the expected results are discovered, review the predetermined results to ensure the results are correctly stated. Include the following information when recording the results of a test:

- Name and version number of the application or document that was tested
- Identification of the input data used in the test (e.g., reel number or file ID)
- Identification of the hardware and operating systems on which the test was run
- Time, date, and location of the test
- Names, work areas, and phone numbers of personnel involved in the test
- Detailed description of the nature of any deviations from predetermined results that were found during the test
- Identification of the output (e.g., reel number or file ID) in which the deviation was found

- Names, work areas, and phone numbers of development area personnel who were informed about the deviations

- Date the developers were informed about the potential problem

- Name and version number of the application or document that was issued to correct the error

- Date the new version was reissued

Record this information each time a test or retest is performed, and log the information in chronological order to serve as an historical document of the test.

If it is determined that the deviation is an error in the system's software, hardware, or documentation, document the error and notify the appropriate development area by using problem reporting procedures established for the project. Include with the documentation a detailed description and supporting documentation when describing the error. Supporting documentation may include images of the test data before and after the test was executed, series of screen images with narratives showing the exact sequence of events that led up to the error, and actual system reports with errors highlighted.

For large-scale development efforts, design a control system to allow one-to-one traceability from the test data to the predetermined result. Group predetermined results in the same order as the input data to allow for easier output review.

5.1.3 Evaluate Results

Compile the outcome of each individual test documented in the VV&T Plan in the Test Results and Evaluation Report along with the evaluation of test methods and test administration procedures. Record the results of each test execution, and compare the test results to the expected results recorded as part of the test documentation. Prepare the Test Results and Evaluation Report, and make a final determination regarding the readiness of the system. Submit the Test Results and Evaluation Report for approval. If the system is approved for production, formally record the decision and certify the system for release.

Compile Test Results

As system acceptance testing continues, maintain information on the execution of each test and the results of all tests in a central repository for the project. Once the execution of all tests and retests is complete, use this information to form the core of the Test Results and Evaluation Report.

Prepare Test Results and Evaluation Report

At the completion of system acceptance testing, prepare a Test Results and Evaluation Report to describe the test procedures and scenarios and the results that were found following their execution. Prepare the Test Results and Evaluation Report following HUD SDM documentation standards (Handbook 2400.15). To prepare the report,

- ***Describe purpose of report***. Identify the project, by name and number, for which the Test Results and Evaluation Report was developed. Document all of the purposes of the report. At a minimum, every Test Results and Evaluation Report will contain the purposes listed below:

  – To document the results of all system acceptance tests

  – To identify deviations from the VV&T Plan

  – To assist in assigning responsibility for resolving issues raised as a result of the system acceptance testing

  – To report on the system's ability to fulfill its intended objectives

  – To provide a basis for estimation of the project completion time

- ***Provide summary of project references.*** List the high-level requirements the system was supposed to meet upon completion of its development. Provide summary information about the sponsoring organization and those organizations involved in developing, testing, using, and operating the system. Also provide any additional information, such as terms, definitions, and acronyms, that will provide clarification for the reader.

- ***Describe security considerations.*** Provide a detailed description of the security requirements that have been built into the system and verified during system acceptance testing. Identify and describe security issues or weaknesses that were discovered as a result of testing.

- ***Provide test analysis of each function tested.*** Provide summary information for each system acceptance test executed. Identify the test, the purpose of the test, the function or capability demonstrated by the test, an analysis of the system’s ability to meet the requirements of the test, and any deviations from the original VV&T Plan that occurred during test execution.

- ***Describe system deficiencies.*** Describe the deficiencies that remain after the system test is completed. Provide traceability to the problem report that was written when the deficiency was detected, and provide the status of the deficiency.

- ***Recommend improvements as required.*** Provide a detailed description of any recommendation discovered during testing that could improve the system, its performance, or its related procedures. If additional functionality is seen as a potential improvement for the user, although not specified in the FRD, it should be included here. Provide a priority ranking of each recommended improvement relative to all suggested improvements for the system.

- ***Summarize capability of the system to meet requirements.*** Provide a summary of the state of the system upon completion of system acceptance testing. This summary should support one of the following test recommendations:

  – The system is virtually error free and should be released into production.

Return to Title Page

- Errors still exist that should be addressed, but a decision could be made to fix these errors in production and not delay release of the system.
- The system has major shortcomings and should not be released into production at this time; instead, it should be returned for further development and retesting.

5.1.4 Perform Configuration Audits

The objective of a configuration audit is to assess whether a system, subsystem, or configuration item meets its technical requirements and if any unauthorized changes are scheduled for the delivery. There are two types of configuration audits: the functional configuration audit (FCA), and the physical configuration audit (PCA).

After acceptance testing for a CI release, perform an FCA to determine if the test results demonstrate that the CI meets its allocated requirements. Perform a PCA to determine if the CI's documentation is complete and consistent with the “as-built” CI.

If a configuration audit uncovers any deviations or discrepancies, or results in action items, the audit leader prepares an action item list and audit report. A satisfactory corrective action plan or deviation authorization must be prepared by the project manager and approved by the user and sponsoring organization before the product can be certified and delivered.

5.1.5 Determine Readiness

Certify the system based on its fitness for use in a production environment as determined by review and approval of the Test Results and Evaluation Report and the physical and functional configuration audits. If the developed system is certified for production, release it.

Certify System

Prepare a certification as to the readiness of the system to be released into the production environment. Summarize the current state of the system. Include the name and version numbers of the system software, hardware, and related documentation and guides that are covered under the certification. Certify the system under one of the following categories:

- The system is virtually error free and should be released into production.
- Errors still exist that should be addressed, but a decision could be made to fix these errors in production and not delay release of the system.
- The system has major shortcomings and should not be released into production at this time; instead, it should be returned for further development and retesting.

Issue Release Request

After certification, issue a Release Request to Computer Services for approval to install the system in the production environment.

5.2 FINALIZE INSTALLATION AND CONVERSION PLAN

Complete the strategies and procedures for system installation and data conversion. Document in the installation strategy the procedures for installing the system application software, support software, security software and hardware, telecommunications equipment, and peripheral devices at all sites. Indicate in the conversion strategy the procedures for converting existing automated and manual files and for ensuring the correctness of the data after conversion. The appropriate organizations within HUD then review and approve the Installation and Conversion Plan. Include the following activities:

5.2.1 Finalize installation and conversion procedures.

5.2.2 Document the finalized plan.

Table 5-4 defines roles and responsibilities of key personnel for finalizing the Installation and Conversion Plan.

Table 5-4. Roles and Responsibilities for Finalizing Installation and Conversion Plan

Role	Responsibility
Project Sponsor	Provides input to the Installation and Conversion Plan.
User	Finalizes the Installation and Conversion Plan.

5.2.1 Finalize Installation and Conversion Procedures

Complete development of strategies for installing the system hardware and software in the production environment and for converting existing data to the format required by the new system. Update the Installation and Conversion Plan to reflect the final strategy, and submit the plan for review and approval by the appropriate organizations before execution of the Installation and Conversion Plan.

Finalize System Installation and Conversion Procedures

Complete the final updates to the strategies for installing and converting the system to the operational environment begun during the Build System phase of development. Include any additional information discovered during the Evaluate System phase of development that may have an impact on these procedures. This is the final update of the plan before data conversion and hardware and software installation. In the procedures, include information on the following:

- Additional climate control
- Wiring and power conditioning
- Physical security and access controls
- Floor plan
- Required supplies

- Hardware and peripheral equipment

Also, finalize decisions on the following:

- Organizations involved and how each organization is involved
- Procedures for installing the system for each site and the sequence of site installations
- Sequence and schedule for the complete (or incremental) installation of the system
- Identification of organizations involved with system installation, each organization's role, and methods (e.g., meetings and communications) for keeping each organization informed about the status of the installation

Account for the following in data conversion procedures:

- Converting existing automated files
- Converting existing manual files
- Staffing requirements and organization for conversion
- Checking accuracy of converted data
- Schedule for data conversion
- Organizations involved with data conversion effort and each organization's role in the effort
- Methods (e.g., meetings and communications) for keeping all organizations informed about the status of the conversion

5.2.2 Document Finalized Plan

Complete the Installation and Conversion Plan to include any updates as outlined above. Ensure that the finalized Installation and Conversion Plan has been developed to meet all applicable HUD SDM documentation standards (Handbook 2400.15). Also ensure that the plan complies with IT procedures appropriate for the system's target platform.

5.3 UPDATE SYSTEM AUDIT STRATEGY

Notify the IG staff about any work products completed during the Evaluate System phase. Inform them of any changes that may have an impact on internal data processing controls as they were described in the Evaluate System phase documentation. Activities include coordinating with IG staff, providing input to IG staff, and updating the Internal Audit Plan.

Table 5-5 defines roles and responsibilities of key personnel for updating system audit strategy.

Return to Title Page

Table 5-5. Roles and Responsibilities for Updating System Audit Strategy

Role	Responsibility
Project Manager	Provides project information to IG staff, as requested.
IG Staff	Coordinates with project manager, as necessary. Updates internal audit plan.

Activities include the following:

- ***Coordinate with IG staff.*** Schedule meetings with the IG staff to inform them of any developments that occurred during the Evaluate System phase. Determine the scope of the information required from the project manager so the IG staff can perform the necessary updates to the Internal Audit Plan.

- ***Provide input to IG staff.*** Provide the IG staff the necessary information.

- ***Update Internal Audit Plan.*** Update the Internal Audit Plan to include information provided by the project manager.

5.4 UPDATE SYSTEM DECISION PAPER

Update the System Decision Paper to include a summary of the progress made during the Evaluate System phase and the schedule of events for Operate System phase activities. Include the project's current status. List in detail any changes to the original plan for developing the project and the parties responsible for approving those changes. Submit the updated System Decision Paper for review and approval by the appropriate management officials. Include the following activities:

5.4.1 Summarize progress of system.

5.4.2 Identify changes.

5.4.3 Summarize schedule of events.

5.4.4 Document results.

Table 5-6 defines roles and responsibilities of key personnel for updating the System Decision Paper.

Table 5-6. Roles and Responsibilities for Updating the System Decision Paper

Role	Responsibility
Project Manager	Updates System Decision Paper, as necessary.
Developer	Provides input to System Decision Paper.
Quality Assurance	Reviews System Decision Paper for adherence to standards.

Return to Title Page

5.4.1 Summarize Progress of System

Summarize the progress of project development through the Evaluate System phase, and update the System Decision Paper accordingly.

5.4.2 Identify Changes

Document any changes to the system development milestones and schedule or any modifications made to the project strategy, and update the System Decision Paper accordingly.

Describe Changes in Milestones, Schedules, and Tasks

Provide a description of any Evaluate System phase occurrences that may have had an impact on the projected milestones, schedules, and tasks of the development effort. Provide a brief explanation of the nature of these impacts, and describe any changes to the milestones or schedule that may occur as a result of these changes.

Describe Modifications to Project Strategy

Provide a description of any modifications to the project strategy that have occurred during the Evaluate System phase of development. Provide a brief explanation of these modifications to indicate the reasons for the changes and any anticipated impacts.

5.4.3 Summarize Schedule of Events

Develop a brief summary of the events that took place during this phase of development and a description of the remaining schedule of development activities. Show the impact these events have had on the original schedule and any anticipated impacts on future development activities.

Describe the portion of the project schedule completed through the Evaluate System phase. Also, describe the schedule of remaining activities planned for the project.

5.4.4 Document Results

Document all information for this phase in the System Decision Paper in accordance with HUD SDM documentation standards (Handbook 2400.15).

5.5 REFINE PROJECT PLAN

The Project Manager controls the project by monitoring phase activities, taking corrective action where necessary; refining the project plan to account for changes attributable to actions taken in the current phase and new information for upcoming phases; and reviewing the quality planning process developed by users, developers, testers, and quality assurance for the next phase. Activities include the following:

5.5.1 Update Project Plan.

5.5.2 Review planned quality process.

Table 5-7 defines roles and responsibilities of key personnel for updating the Project Plan.

Table 5-7. Roles and Responsibilities for Refining Project Plan

Role	Responsibility
Project Sponsor	Approves changes to project schedule. Approves changes to Project Plan.
Project Manager	Updates Project Plan to include actual costs incurred and specific activities accomplished for the Evaluate System phase; and revises project schedule, plans, strategies, resources, and requirements for the Operate System phase, as required. Reviews, for approval, the quality process planned by users and developers for the next phase.
User	Provides input to project manager on required Operate System phase tasks and activities. Attends review of Project Plan to provide input to the review process as required. Works with developer to review and update the quality process planned for the next phase.
Developer	Provides input to project manager on Required Operate System phase tasks and activities. Participates in review of Project Plan. Works with user to review and update the quality process planned for the next phase.

5.5.1 Update Project Plan

Update the plan with cost, schedule, and budget data for the current phase (Evaluate System) to the level of detail necessary to reflect the project's status. Review the plan and determine if any activities, described for the next phase (Operate System), are affected by the completion status of the current phase activities. Adjust schedules and resource requirements for activities in the next phase, if necessary, and assign starting and ending dates for affected activities. Take into account that the starting date for some activities may depend on the completion of other activities. Update milestones, schedules, and resource requirements for the remainder of the project.

5.5.2 Review Planned Quality Process

Users, developers, testers, and QA work closely together to determine the process to be used to build the product with the desired quality. Activities include tailoring the planned quality process for the next phase and identifying the standards and procedures to be used. The project manager reviews the quality process for approval and execution in the next phase.

5.6 REVIEW AND APPROVE DOCUMENTS AND DELIVERABLES

Review and approval of documents and deliverables is an iterative process as each phase provides more definition and products are revised. In addition, the approval process offers senior management the capability to monitor the project. Activities include the following:

5.6.1 Review revised products from prior phases, as required.

5.6.2 Submit Evaluate System phase documents for review.

5.6.3 Obtain approval of technical and documentation deliverables.

Table 5-8 defines roles and responsibilities of key personnel for reviewing and approving documents and deliverables.

Table 5-8. Roles and Responsibilities for Review and Approval of Deliverables and Documents

Role	Responsibility
Project Sponsor	Participates in project review (optional).
Project Manager	Determines needed changes to prior phase documents and ensures revision in accordance with the overall Project Plan. Coordinates review of Evaluate System phase deliverables. Determines level of required review and schedules required reviews. Attends reviews and presents deliverables. Ensures that Evaluate System phase deliverables are updated to include any recommendations received during the review. Obtains appropriate concurrences and approvals for Evaluate System phase document deliverables and associated management summaries. Obtains appropriate concurrences and approvals for revised documentation.
User	Revises prior phase products when applicable. Assists the project manager with the preparation of Evaluate System phase deliverables for review. Attends deliverable reviews.
Developer	Revises prior phase products when applicable. Assists the project manager with preparation of Evaluate System phase deliverables for the review. Provides technical expertise during the review process.
System Acceptance Test Team	Provides technical expertise during the review process.
ADP Security	Reviews Evaluate System phase deliverables to ensure that all necessary security requirements have been thoroughly tested. Reviews revised products to ensure that the necessary security standards and guidelines have been addressed.
Computer Services	Participates in review of documents and deliverables.
Configuration Management	Assists project manager with review preparation.
Quality Assurance	Ensures that all review procedures are followed as required. Reviews Evaluate System phase deliverables to ensure they meet all applicable HUD standards and guidelines.

5.6.1 Review Revised Documents from Prior Phases as Required

Review the activities performed during the Evaluate System phase to determine if they have an impact on any documents produced during the previous phases. If changes are required to prior lifecycle phase documents, update these documents to reflect current project developments. Project personnel review revised documents in a manner similar to reviews for current phase documents to ensure the changes are within the scope of the project's requirements and are in compliance with HUD and project standards and procedures. For each revised document, prepare a management summary that summarizes the essential revisions. Submit the revised products to the appropriate review board for approval. Use the appropriate document review checklist in Appendix E to assist in the reviews. File affected product change records with CM for appropriate version control updates.

5.6.2 Conduct Review for Evaluate System Phase Documents and Deliverables

Conduct a project review with project personnel and system stakeholders to ensure that the project documents and deliverables for the Evaluate System phase include the necessary level of detail, fulfill the system's requirements, and meet the appropriate HUD and project standards and guidelines. Use the appropriate document review checklists in Appendix E for Evaluate System phase deliverables to aid in the document reviews.

A minimum of 10 working days before the scheduled review, notify the personnel required to attend and provide each with a copy of the product for pre-review. Discuss all comments or objections raised during the review, and reach a consensus on one of the following before the review session terminates:

- The documents are correct and complete, as is, without any further changes.
- Additional changes that need to be made are minor and do not require further review. In this case, the updates should be made and change pages should be distributed by an agreed upon date.
- Required changes will have a major impact on the plan. A second review must be scheduled. The changes must be incorporated and the resulting change pages distributed a minimum of 10 working days before the second review.

Prepare a management summary for each document that includes the essential data collected in the document, lists conclusions that may be drawn from the document, and describes the potential impacts on the project, if applicable. Submit the management summaries to the appropriate review board for approval. Documents for approval include the following:

- Test Results and Evaluation Report
- Installation and Conversion Plan
- System Decision Paper (updated)

5.6.2 Obtain Approval of Project Documents and Deliverables

Present the project documents and deliverables to the chairperson of the appropriate review board for approval at least 10 days before the scheduled decision date. Include the management summary information, approval (sign-off) record, the System Decision Paper, and any other required or requested information. The review board chairperson coordinates review board comments, recommendations, and approval signature(s), and returns the approval record to the project manager. Recommendations on the approval record are addressed by the project, and the document should be resubmitted to the board if requested. Approval will be assumed if there is no response or if the response is "no comment." Project approval records are maintained by the project's configuration management (CM) function. A copy is inserted into the central project library.

The project proceeds to the next phase after all project documents and deliverables are approved.

5.7 PERFORM CHANGE CONTROL ACTIVITIES

Any products developed or procured during the Evaluate System phase are baselined and subjected to version control. Products baselined during prior activities are assigned new version control numbers when they undergo change (e.g., an update or rewrite). CM reports are provided to the Project Manager as requested or required.

Table 5-9 defines roles and responsibilities of key personnel for performing change control activities.

Table 5-9. Roles and Responsibilities for Performing Change Control Activities

Role	Responsibility
Project Manager	Approves controlled products for distribution.
Configuration Management	Performs change control for new and revised products. Provides status reports to project management as requested or required.
Quality Assurance	Audits products to ensure only approved changes are addressed.

The CM function prepares the baseline for configuration control. The baseline includes any new technical and document deliverables that will comprise the project configuration baseline. For the baseline items, CM performs the following activities:

- ***Verify changes made to product***. Review the updated product to ensure that the changes have been made as described in supporting documentation. Supporting documentation may be comments received from document reviews. Supporting documentation for software may be software change requests or discrepancy reports generated during testing.

- ***Assign version number***. The version number must follow conventions established by HUD and enable the project's CM to monitor updates to the product and assist in its distribution. For software (CIs) programs, increment the version number each time a change is made to the software to correct deficiencies found during testing.

- ***Store approved version in central library***. After each baselined product has been completed and approved, according to HUD procedures, store the approved version in the project's central library. The project's CM function controls access to the library. For software, the library can be a subdirectory or dataset where all baselined software CIs will be stored.

- ***Record product information in inventory log***. Maintain an inventory log that includes the title of the product, release date, version number, name and version or model numbers of the software and hardware used in the development of the product, name of the organization responsible for development of the product (usually the sponsoring organization), and the product distribution list.

- ***Distribute copies of products as required***. The project's CM function distributes copies of the products according to a distribution list maintained as part of the inventory log information, with distribution based on need and the security level of the product

- ***Archive old versions of products***. Archive and retain outdated versions of all products for the required period of time, in keeping with HUD standards and guidelines.

This page intentionally left blank.

Return to Title Page

Section 6. Operate System

Return to Title Page

6.0 Operate System

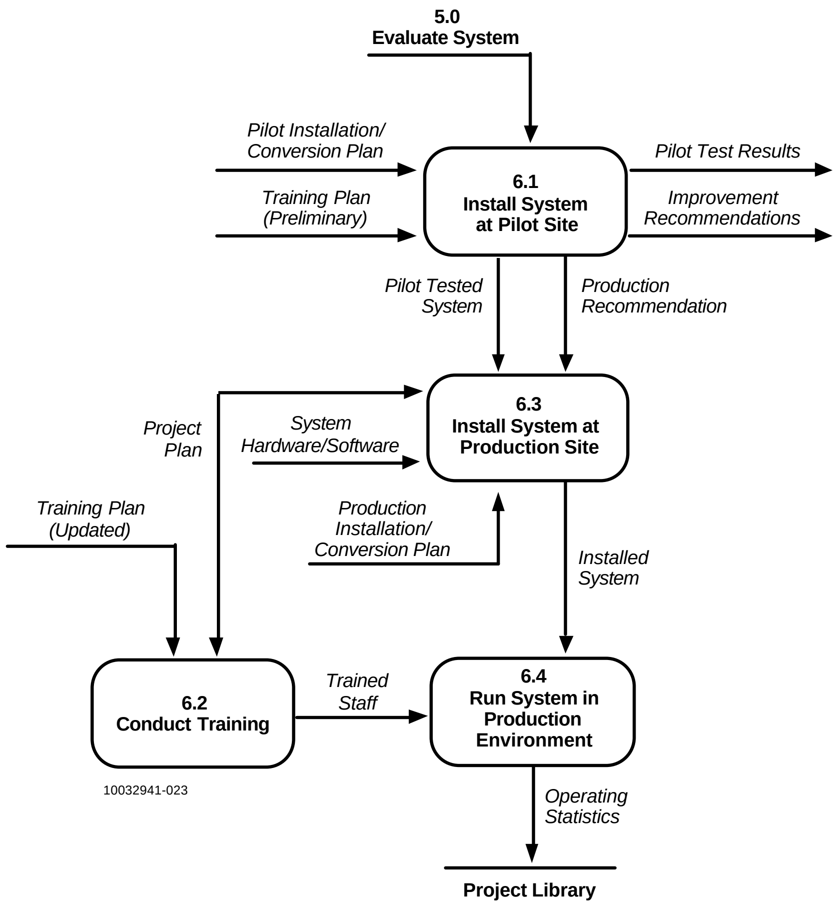

Figure 6-1. Process Flow for Operate System Phase

Return to Title Page

6.0 OPERATE SYSTEM

Purpose

The purpose of the Operate System phase is to put into production the certified system. The system is initially installed at a pilot site and eventually released into its full-scale production environment during the Operate System phase. All necessary training for using the system is accomplished, and the system's performance is monitored in a production environment. Figure 6-1 highlights the process flow for the Operate System phase.

Overview

Operate System is the final phase of the development lifecycle, when the system is released, initially to a pilot site and then into the full-scale production environment. The piloting of the system at a specified pilot site ensures that the system will perform all of its functions to meet full-scale operational requirements and serves as the vehicle for verifying that the system is acceptable to the users. The minimum time for piloting a system should be long enough to complete one full processing cycle.

During this phase, supporting organizations perform the following activities:

- Install the system at the pilot site.
- Upon completion of the pilot tests, release the system into operation at all sites originally specified in the project plan.
- Perform installation and conversion procedures at both the pilot and operation sites in accordance with the Installation and Conversion Plan.
- Hold training classes for user and operations personnel in accordance with the Training Plan. Complete the training modules, acquire the necessary training resources, schedule the training sessions, and notify the system users about the schedule.
- Perform project peer reviews of deliverables listed in Table 6-1.
- Upon request, present the System Decision Paper and all required supporting documentation to the appropriate review boards for approval.

Table 6-1. Operate System Phase Functions and Products

Operate System Functions	Products
6.1 Install System at Pilot Site	Pilot system Pilot Test Report
6.2 Conduct Training	Training material (updated)
6.3 Install System at Production Site	Production system
6.4 Run System in Production Environment	Project Plan (Complete)

Return to Title Page

Standards and Guidelines

HUD SDM documentation standards (Handbook 2400.15) and project guidelines should be followed during the Operate System phase functions.

Roles and Responsibilities

Throughout the Operate System phase, key personnel are required to perform various activities. Table 6-2 lists the types of personnel required and the activities for which they are responsible.

Table 6-2. Roles and Responsibilities for Operate System Phase (1 of 2)

Roles	Responsibility
Project Sponsor	Approves project schedule revision, if necessary. Participates in review. Formally accepts pilot system for production use.
Project Manager	Revises project schedule, if necessary. Coordinates system installation activities. Coordinates the return of resources and the closing of the project. Presents pilot test results for user approval.
Project Development Team (User and Developer)	Produces necessary training materials. Schedules training classes. Finalizes training materials and schedules, and conducts training classes. Evaluates training classes and modifies them, as necessary. Updates system documentation. Installs system. Monitors system performance to ensure responsiveness to user's needs. Modifies system software, as necessary. Assists in readying pilot environment. Executes pilot Installation and Conversion Plan. Assists in readying production environment. Makes modifications and enhancements to system hardware, as necessary.
Computer Services	Ensures that pilot environment is ready. Performs pilot system installation and conversion. Performs day-to-day operations of system. Performs system installation(s). Ensures that production environment is ready. Performs production system installation and conversion. Processes problem tracking and resolution. Monitors system performance to ensure responsiveness to user community.

Table 6-2. Roles and Responsibilities for Operate System Phase (2 of 2)

Roles	Responsibility
ADP Security	Conducts periodic security reviews on the system when in production.
Project Database Administrator	Supports data conversion activities.
Configuration Management	Performs change control activities. Establishes and maintains production baseline.
Quality Assurance	Monitors system installation and conversion activities. Audits controlled production baseline.

6.1 INSTALL SYSTEM AT PILOT SITE

Prepare the pilot site and install and/or convert the hardware, software, and databases by following the Installation and Conversion Plan and the HUD ADP Standard Release Procedures. Closely monitor the system during operation at the pilot site and document the results. Based on the documented results, make recommendations to the user and project sponsor organizations about the system's operational readiness in the production environment. Include the following activities:

6.1.1 Ensure pilot environment is correctly established.

6.1.2 Execute Installation and Conversion Plan.

6.1.3 Conduct pilot site training.

6.1.4 Operate system in a pilot environment.

6.1.5 Document results and make recommendations.

Table 6-3 lists personnel roles and responsibilities for installing the system at the pilot site.

Table 6-3. Roles and Responsibilities for Installing System at Pilot Site (1 of 2)

Role	Responsibility
Project Sponsor	Reviews pilot test results and accepts pilot system for production.
Project Manager	Coordinates system installation activities. Presents pilot test results to system users.
User	Operates pilot system.
Developer	Installs system. Train pilot system users. Assists in readying pilot environment. Executes pilot Installation and Conversion Plan. Updates system documentation.
Computer Services	Ensures pilot environment is ready. Performs pilot system installation and conversion.
Project Database Administrator	Supports data conversion activities.

Return to Title Page

Table 6-3. Roles and Responsibilities for Installing System at Pilot Site (2 of 2)

Role	Responsibility
Configuration Management	Establishes controlled pilot environment. Establishes production baseline. Logs changes implemented during pilot activities. Controls approved system documents.
Quality Assurance	Monitors pilot site installation and conversion activities. Audits production baseline library. Verifies problem/change logs.

6.1.1 Ensure Pilot Environment Is Correctly Established

Review all equipment and site facilities to ensure that the pilot operation environment is ready for the newly developed hardware and software to be installed. Notify affected personnel and organizations about the upcoming installation and conversion, and schedule meetings to ensure that all affected personnel are aware of any procedural changes.

Ensure That Physical Facilities Are Ready

At the pilot site, perform a review of the equipment and the physical environment where the equipment will be located to ensure that all safety regulations have been followed and that the installation and conversion procedures were carried out in accordance with the Installation and Conversion Plan.

Ensure That Affected Organizations Are Notified

Contact the organizations that will be affected by the pilot of the new system to ensure they are aware of the cutover date for the pilot program. Make sure that these organizations are aware of their responsibilities and of any revised operating procedures required during pilot activities, that they have completed the necessary preparations, and that they are ready for the pilot to begin.

6.1.2 Execute Installation and Conversion Plan

Install the hardware and software for the new system at the pilot site in accordance with the Installation and Conversion Plan. Convert or install the necessary data and databases and initiate system performance monitoring functions.

Execute System Installation in Accordance with Plan

Install the system at the physical location where the pilot test is to be performed. Ensure that the installation is carried out in accordance with the procedures described in the Installation and Conversion Plan. Verify that only the required hardware and software are installed in the pilot test environment. Conduct initial test checkout to verify correct integration of the installed components.

Execute Data Conversion in Accordance with Plan

Convert the data to be used in the pilot system to the media and format required by the new system. Ensure that data conversion is carried out in accordance with the procedures described in the Installation and Conversion Plan. Verify, by inspection or test, that data conversion has been implemented correctly. If problems are encountered during data conversion, follow the problem resolution procedures defined in the Installation and Conversion Plan.

Control Pilot Environment

Ensure that the integrity of the system is preserved during installation. Review the list of equipment to ensure that all components necessary for the pilot site are installed. Prepare software installation package from the configuration controlled software library. Maintain all records associated with the installation (e.g., problem reports, audit reports, and installation reports).

Pilot Readiness Review

When installation is complete, notify the project sponsor and user organizations that piloting activities are ready to begin. Be prepared to discuss the steps of the installation process, the equipment and software installed, and tests to be performed to show the users that the system is ready to be used in the pilot environment.

6.1.3 Conduct Pilot Site Training

Schedule and conduct training classes for all personnel affected at the pilot site. Perform pilot site training in accordance with the Training Plan. Monitor training activities to determine if the selected training techniques are achieving the desired results.

Analyze feedback received from personnel attending the pilot training sessions; and, based on this analysis, refine training procedures or materials to ensure that training objectives are met. Changes that occur during pilot system activities may require changing the Training Plan and related courses and materials.

6.1.4 Operate System in Pilot Environment

Operate the newly installed system in the pilot environment designed to mirror all aspects of the production environment. Closely monitor all aspects of system performance for compliance with relevant system documentation, and report and correct any deviations before a full-scale production release. Pilot operation should be run parallel to the existing system(s) so that the data output from the two systems can be compared to ensure that the new system is operating correctly.

Perform Day-to-Day Operation of Pilot System

During day-to-day operations at the pilot site, follow the procedures described in the User’s and Operations Manual. Ensure the following:

- The system performs the transactions or functions for which it is designed.
- System performance meets or exceeds required capabilities.
- Necessary computer time has been scheduled for the systems.

- Resources necessary for the system to run correctly are available (e.g., disk space, tapes, and computer paper).

- Personnel are available to run the pilot system in parallel with the existing systems.

Maintain System Logs

Ensure that operations personnel maintain proper system logs that meet the requirements of operations. These system logs identify any problems found or changes implemented during pilot activities. The system logs are to be used to verify that all problems have been identified, corrected, or deferred to a future system release. Use existing HUD automated tools for maintaining system logs.

Maintain and Follow System Documentation

Ensure that all changes made to the system in the pilot environment are updated in the system documentation. Use configuration version control procedures for all system documentation throughout the life of the system, including pilot operation.

Resolve Problems Found While Operating Pilot System

If error messages are received during the pilot operation of the system, document them as required, and follow the response procedures as outlined in the User’s Manual or Operations Manual. If the response procedures do not have the desired effect or if the error message is not listed in either guide, contact the system administrator for support in analyzing the cause of the message.

Report all errors detected during the pilot operation of the system to the project's development staff by using the agreed-upon discrepancy reporting procedures. Report problems determined to have an impact on system security to the ADP security staff.

6.1.5 Document Results and Make Recommendations

Document the results of pilot system operation in a Pilot Test Report. Use the system log to record any errors detected during the pilot operation and all feedback received from monitoring the system. Closely review the results and the actions taken to resolve errors, and develop and document any recommendations for improving system performance. Submit these recommendations to the appropriate management areas for consideration.

Document System Problems

Document problems that surface as a result of system diagnostic checks, system performance tests, user or operations personnel complaints, or detected errors. At a minimum, this documentation should include the following:

- List of any error messages that may have been received either at the terminal or at the operator’s console

- Name and number of the application that was in use at the time of the problem

- List of the data being input to the application when the problem occurred

Return to Title Page

- Time and date of the occurrence
- Detailed description of the nature of the problem
- Function being performed and segment of the system in use at the time of the error

Document Results of System Operation

A continual record should be kept of the results of the system's performance in its pilot operational environment. At a minimum, include the following in this documentation:

- Time and date that the system was in use
- Types of applications that were run on the system, if applicable
- Number of users during various predefined times during the day or night
- Problems that may have been encountered
- Detailed description of any measurable degradation in response time

This documentation can usually be created by automated techniques, such as using the operating system's ability to dump information concerning CPU time, disk input and output, number of lines printed, elapsed time, and number of jobs or steps processed.

Approve System

System approval is the ultimate objective of the system development project. At the completion of the pilot operation, review pilot test results with the project sponsor and user organizations. Highlight project requirements and functions performed in the pilot environment, results of pilot operation, and any outstanding system problems. Upon successful review of the test results, the user should formally sign off for acceptance of the system before the system is released into production.

Establish Production Baseline

Upon system approval, the production environment is baselined. Maintain strict version and change control over this baseline. Conduct periodic audits to verify that only approved requirements and changes are incorporated into the baseline. Use automated CM tools for processing approved changes and generating change status reports on demand.

6.2 CONDUCT TRAINING

Finalize training materials, and schedule training classes for all affected personnel. Conduct all training sessions in accordance with the Training Plan, and monitor training activities to determine if the selected training techniques are achieving the desired results. Include the following activities:

6.2.1 Finalize training materials.

Return to Title Page

6.2.2 Finalize training schedule.

6.2.3 Conduct training sessions.

6.2.4 Evaluate effectiveness of training.

6.2.5 Modify training materials as necessary.

Table 6-4 lists personnel roles and responsibilities for conducting training.

Table 6-4. Roles and Responsibilities for Conducting Training

Role	Responsibility
Project Manager	Finalizes training materials. Schedules training sessions.
Developer	Conducts training sessions.
User	Attends training sessions.
Quality Assurance	Monitors training sessions.

6.2.1 Finalize Training Materials

Complete all materials necessary to support the scheduled training, including training booklets, online exercises, and presentation materials. Review these materials to ensure compliance with the project Training Plan. Modify training materials to reflect any corrections or changes to the system that may have resulted from deficiencies found during systems acceptance and pilot tests.

6.2.2 Finalize Training Schedule

Complete scheduling of training sessions for all required personnel. Develop the training rosters and attendance sheets for each session. Notify all personnel to be trained about the time and location of their respective sessions. Ensure that the training facilities and equipment have been reserved in advance.

6.2.3 Conduct Training Sessions

Carry out the approved training sessions in accordance with the project Training Plan. At the completion of each training session, elicit feedback from personnel to ensure training objectives were met.

6.2.4 Evaluate Effectiveness of Training

Analyze all feedback received from personnel attending training sessions; and, based on this analysis, make recommendations for changes to training procedures or materials to ensure that training objectives are met, as described in the project Training Plan.

6.2.5 Modify Training Materials as Necessary

Respond to recommendations made as a result of analyzing feedback. Update or change training materials and course procedures to ensure compliance with all approved recommendations received. Changes to the system itself also may require updating the Training Plan and related courses and materials.

6.3 INSTALL SYSTEM AT PRODUCTION SITE

Perform inspections of all designated production sites. Detect and correct deficiencies in the production environment before installation of the new system. Perform all system installation and conversion procedures in compliance with the Installation and Conversion Plan. Include the following activities:

6.3.1 Ensure production environment is correctly established.

6.3.2 Execute Installation and Conversion Plan.

Table 6-5 lists personnel roles and responsibilities for installing system at production site.

Table 6-5. Roles and Responsibilities for Installing System at Production Site

Role	Responsibility
Project Manager	Coordinates system installation activities.
Developer	Assists in readying production environment. Executes installation plan.
Computer Services	Ensures the production environment is ready. Performs production system installation and conversion.
Quality Assurance	Monitors system installation and conversion activities.

6.3.1 Ensure Production Environment Is Correctly Established

Perform facility inspections of all areas designated as production sites for the new system. Document and correct any deficiencies detected during these inspections before the installation and conversion of the system at the production sites. Notify all personnel affected by the new system about the upcoming installation and conversion, and perform a review to ensure that all necessary staffing and training are complete.

Ensure That Physical Facilities Are Ready

Perform a review of the equipment and the physical environment where the equipment will be located to ensure that all safety regulations have been followed and that the installation and conversion procedures have been carried out in accordance with the Installation and Conversion Plan.

Ensure That Affected Organizations Are Notified

Contact the organizations that will be affected by operation of the new system to ensure they are aware of the cutover date for the production release. Make sure that they are aware of their responsibilities and any revised operating procedures affected by the new system, that they have completed the necessary preparations, and that they are ready for the system to become operational.

Return to Title Page

6.3.2 Execute Installation and Conversion Plan

Complete installation of system, hardware, and software. Complete all conversions of data and databases. Closely monitor installation and conversion procedures to ensure compliance with the project's Installation and Conversion Plan and the HUD ADP Standard Release Procedures.

Execute System Installation in Accordance with Plan

At each site where it is to be operated, install the system in production mode and establish user password and security authorizations when required. Ensure that the installation is carried out in accordance with the procedures described in the Installation and Conversion Plan. Include installation procedures for all system components, including hardware, software, networks, and workstations. At the completion of the system installation at each site, obtain a final signoff to signify that the system has been thoroughly checked.

Execute Data Conversion in Accordance with Plan

Convert the data to be used by the system, when it is operational, to a media and format acceptable to the new system. This conversion should be performed following the procedures described in the Installation and Conversion Plan, including the following:

- Placing data files and their contents on the system
- Checking converted data with source data to ensure correctness
- Determining source data retention requirements should any problems arise during operation of the system

6.4 RUN SYSTEM IN PRODUCTION ENVIRONMENT

Schedule production runs for the new system. Perform all daily activities necessary to operate the system, including monitoring the system’s performance to ensure adequate response time, system security, and problem-free operation. Take actions to correct any problems uncovered during system monitoring. Include the following activities:

6.4.1 Schedule production runs.

6.4.2 Monitor system performance.

6.4.3 Ensure system is responsive to users.

6.4.4 Ensure system is consistently available.

6.4.5 Report discrepancies for problem resolution.

6.4.6 Determine potential system modifications and enhancements.

6.4.7 Upgrade system as required.

6.4.8 Ensure security of environment.

6.4.9 Revise resource plan.

Return to Title Page

6.4.10 Close the development project.

Table 6-6 lists personnel roles and responsibilities for establishing a production environment.

Table 6-6. Roles and Responsibilities for Establishing Production Environment

Role	Responsibility
Project Manager	Monitors system performance activities. Closes out development project.
Developer	Monitors system performance to ensure responsiveness to users' needs.
ADP Security	Conducts periodic security reviews on production system.
Configuration Management	Retains records in project library.
Computer Services	Monitors system performance to ensure responsiveness to user community. Performs daily operation functions.

6.4.1 Schedule Production Runs

Schedule and execute the new system's runs in a production environment, and carry out all production personnel responsibilities in accordance with the requirements described in the Operations Manual. Ensure that records produced by the system are retained in the project library.

6.4.2 Monitor System Performance

Monitor the daily operation of the system to ensure the following:

- Capacity of the platform is not exceeded.
- System is operating with sufficient response time.
- Sufficient transaction throughput occurs.
- Downtime is monitored (mean time between failures [MTBF] and mean time to failure [MTTF]).
- System is operating efficiently.

6.4.3 Ensure System Is Responsive to Users

Perform periodic reviews to assess responsiveness of the system. Address all facets of the system, including response time and throughput, user satisfaction with functional and data requirements, technical performance, and system management. Document recommendations to ensure that the system continues to respond adequately to users' needs.

Conduct Review of System

Schedule and conduct periodic surveys of the system's users to determine their level of satisfaction with the current operation.

Identify Any New Requirements

Describe any new requirements that may be imposed on the system. Ensure that the description is of sufficient detail to enable maintenance programmers to make the necessary modifications.

Formulate Recommendations

Provide recommendations for updating the system to meet the new requirements or improve performance. Ensure that recommendations for change include changes to system documentation and to software and hardware.

Document Findings

Describe the recommended changes by providing as much detail as is available for developers to act on the recommendations. Document these recommendations and adhere to the applicable HUD standards and guidelines.

6.4.4 Ensure System Is Consistently Available

While the system is operational, ensure that all functions of the system are available to all users, as required, and that the users are interfacing with an error-free system. Measure availability based on the percentage of total time that the system is available to users.

Schedule for Maximum Use of System Resources

Ensure that operations personnel are aware of the user’s requirements and that the operations schedule allows the user access to the system when needed. Schedule all activities that may result in an interruption of service to the user (e.g., system backup and updates) during the hours that will have the least impact on the user’s needs.

Reduce Downtime for Performing Maintenance

Although periodic maintenance is required to keep the system responsive to the user’s needs, perform such maintenance in a manner that will reduce the amount of time the system will be down. This may mean consolidating system updates so that they can be performed at one time, thus reducing the number of times the system needs to be brought down, or performing maintenance during off hours when the user has no need for the system.

6.4.5 Report Discrepancies for Problem Resolution

If system monitoring reveals areas of performance that need improvement, take the necessary steps to restore system performance to acceptable levels. Use the Service Ticket Action Resolution System (STARS), which provides IT a centrally located repository for logging computer-related requests and problems and provides a facility to track completion of problem resolution. Describe the problem, on a STAR form, in as much detail as possible based on information received during system monitoring. Provide as much information as possible to assist in resolving the problem.

6.4.6 Determine Potential System Modifications and Enhancements

Periodically assess the system to determine potential modifications and enhancements. Use findings and recommendations presented to management from the post-implementation review as input for the assessment. During these assessments, target the following areas:

- Parts of the system that did not meet user expectations or instances in which the implementation of a requirement is inefficient
- Parts of the system that do not function according to specifications
- Potential scheduling or process improvements
- Functions requested by the users as their familiarity with the system increases

6.4.7 Upgrade System as Required

Release upgrades of the system software and hardware into the production environment, when required, to ensure that the system remains responsive to the user's needs. Such upgrades may incorporate improved or additional functions or data, updated system hardware, increased capacity, or improved system performance.

Perform Software Releases and Obtain Release Approval

Perform software releases in accordance with HUD standards and guidelines. Forward all release requests to HUD Computer Services for approval before software is released into production. HUD Computer Services verifies completion of appropriate testing, appropriate user notification, and necessary documentation to support the release. HUD Computer Services then verifies that all necessary preparations for releasing the software and supporting the software in production have been satisfactorily completed and that the release schedule does not conflict with any other data center activities or processing.

Upgrade Hardware in Accordance With Operations Procedures

Perform periodic monitoring to determine if an upgrade or replacement of hardware is required. If a determination is made to replace or upgrade hardware, develop a plan detailing the type of upgrade, the impact on the system, and the date on which the upgrade is required. Before the actual upgrade, Computer Services must approve the upgrade plan. Perform all hardware upgrades or replacements in accordance with Computer Services procedures.

6.4.8 Ensure Security of Environment

Periodically, review the system to ensure that all facets of the system, including hardware, software, communications, personnel, system procedures, standards, and contingency plans, still meet all requirements for certification as outlined in applicable HUD, Office of Management and Budget (OMB), and other Federal policies, regulations, and standards.

Review System

Perform periodic reviews of the system software, hardware, operating environment, and procedures for access to the system environment as well as the system itself (password procedures) to ensure that the system's security controls are still performing as designed. Any changes to HUD security requirements require that additional reviews be performed on the system to ensure that the system meets the new requirements.

Return to Title Page

Document Findings

Prepare a detailed description of the security review that was performed. Include the following information in the description:

- Name of the system that was reviewed
- Names of the individuals that performed the review
- Date and time of the review
- Specific areas of the system that were reviewed (i.e., computer room, password system, and restart capabilities), including version numbers if applicable
- Review findings
- Reviewers’ recommendations

6.4.9 Revise Resource Plan

Revise the project resource plan to reflect the reduction of development personnel required by the project at this point. Identify the personnel required to support maintenance of the system.

Release Personnel

When development is completed and the system is released, nonmaintenance-related personnel should be released from the development project. The most experienced team members should be kept until the development project is physically ended.

Identify Staff for Maintenance

If the project team is to maintain as well as develop the system, identify maintenance personnel for the project. When maintenance efforts are initiated, draw personnel resources from the identified individuals.

6.4.10 Close the Development Project

Before the system is officially turned over to production, develop a shutdown checklist. The checklist should include questions such as the following:

- Has the facilities organization responsible for the space been notified about the date for closing the development project?
- Have all field sites been notified about the person to call for maintenance, enhancements, or changes?
- Has the Internal Audit Plan been reviewed for completeness, and has the final System Decision Paper been delivered to the appropriate management officials?
- Have all project members been placed on new projects or returned to their functional organizations?

Complete the Project Plan with a section that contains software closeout data and characterizes the project effort. Finalize and include the project’s historical records and lessons learned for use by other software development projects.

This page intentionally left blank.

Return to Title Page

Appendix A. Project Management Guidelines

Return to Title Page

This page intentionally left blank.

APPENDIX A. PROJECT MANAGEMENT GUIDELINES

The goals of project management are the following:

- To perform the specified work on schedule and within the negotiated cost
- To deliver products that are of high quality
- To identify problems in schedule, cost, or quality; assess their impact; and select and initiate corrective action
- To communicate to the user, senior management, and other reporting levels an objective assessment of status (including problems, impact, and corrective actions)

Project management uses an iterative process of planning and controlling to meet its goals. Project planning and control consists of detailed activities that represent the overall management methodology (see Table A-1). Although this methodology is applicable to all projects, the project manager may tailor the approach to reflect size, scope, and complexity differences among Information Technology (IT) projects.

During project planning, the project manager is responsible for working closely with the sponsor and user to define the scope of the project, establish a responsive project team, and determine the baselines necessary to control schedule and cost.

During project monitoring and control, the project manager works closely with the sponsor and user to guide all aspects of the work and coordinate key decisions. The manager tracks all key parameters representing costs, schedules, and resources against the established baselines; assesses status; initiates corrective actions; and reports both status and actions in an accurate and timely manner to users, senior management and appropriate reporting levels.

Table A-1. Project Management as a Set of Iterative Activities

Planning	Controlling
Define work. Determine technical approach. Organize work into measurable and manageable packages. Assign project team. Describe roles and responsibilities. Determine resource requirements. Identify special planning needs. Establish process for project reviews. Establish process for document approvals.	Collect, review, and post project data. Calculate variances. Evaluate project status. Resolve variances and address problems. Report project status. Manage change and issues. Manage risk. Manage configuration. Manage quality. Measure productivity and quality. Keep the Project Plan up to date.

A.1 PLAN THE PROJECT

Project management begins with project planning. Project planning involves understanding the scope of work; developing a technical approach; defining, scheduling, and budgeting the work; organizing the project team; and addressing special project considerations and documenting them in the Project Plan. Most activities described in this section are overlapping and iterative in nature; some activities may be conducted in parallel with others. Activities include the following:

- A.1.1 Define work.
- A.1.2 Determine technical approach.
- A.1.3 Organize work to measurable and manageable packages.
- A.1.4 Assign project team.
- A.1.5 Describe roles and responsibilities.
- A.1.6 Determine resource requirements.
- A.1.7 Identify special planning needs.
- A.1.8 Establish process for project reviews.
- A.1.9 Establish process for document approvals.

A.1.1 Define Work

During project planning, the project manager works closely with the user and sponsor to define and understand the scope of the project. Project managers should not wait to receive the final set of system requirements to begin project planning. Project planning should begin as soon as project personnel become aware of the user's intent to initiate a new software effort. Identify all activities planned for development and implementation of the system from coordinated input received from organizational areas that will be involved in the project's lifecycle. Receive the input during the initiation of the project and at the latter part of each lifecycle phase. The organizational areas include developers, users, security, data and database administration, configuration management (CM), and quality assurance (QA).

The project manager records in the Project Plan his or her understanding of the following:

- User's requirements, constraints, and contributions
- User's goals and objectives
- Management's risk tolerance level
- Products and their characteristics to be delivered to the user

Ascertain the user's requirements and constraints from discussions with the user and sponsor. For most development projects, the only basis for requirements is the HUD Information Strategy Plan (ISP). For other lower level development and maintenance projects, the user may have only a general idea of the requirements; determining detailed requirements may follow later or may even be part of subsequent project lifecycle phases.

HUD experience has been that the more the end user is involved throughout the project's lifecycle, the more likely the expected software product will meet the user's needs and expectations. This involvement is particularly crucial during requirements definition and software design activities early in the lifecycle.

During meetings with the user, determine the user's organizational goals and specific project objectives regarding overall cost, schedule, and product quality. Also understand the goals of higher levels of the organization. The project manager and user examine the goals and available budget to determine and agree upon the appropriate schedule for delivery, robustness desired of the product, and formality and level of detail in documents and reviews. When lifecycle time decreases, the user should expect fewer capabilities within a fixed budget. The project manager defines in the Project Plan the specific approach that best addresses the application domain, the objectives established for the project, and the size of the effort. Table A-2 provides examples of the types of objectives that might be established for a project.

Table A-2. Sample Project Objectives

Objective	Examples
Cost	Minimize cost to develop. Minimize cost to maintain.
Schedule	Deliver by fixed date.
Product qualities	Maximize reusability. Maximize robustness. Maximize freedom from defects. Maximize performance (e.g., response time). Maximize maintainability.

Reach agreement with the user regarding products to be delivered and goals for each product with respect to cost, schedule, and quality. ***Remember that initially the user may have only a general idea of product requirements*** and will probably determine specific products later.

Use interim software releases (as opposed to one comprehensive release) either to satisfy the user's early operational needs or as a risk mitigation technique. Early releases also help to build the user's confidence, as well as management's, that the project is on track.

A.1.2 Determine Technical Approach

After the project manager defines a reasonable scope for the work, the next step is to define a technical approach that best accomplishes the work. The project manager defines the project's technical approach by selecting an appropriate lifecycle model for performing required activities and appropriate packaging techniques for required products. To facilitate developing each project's technical approach and to avoid "reinventing the wheel," this section contains a summary of effective lifecycle models from among which the project manager can choose to satisfy the project goals.

After selecting an appropriate lifecycle model, a project manager must determine the activities, methods, techniques, and products that will help achieve the goals and objectives established for the project. The project manager should incorporate into the Project Plan the appropriate project activities and HUD control and reporting mechanisms provided in the SDM.

A lifecycle model is composed of one or more phases (for example, a define phase, a design phase, a build phase). Within each phase, one or more activities is executed. For example, during the waterfall model's design phase, the design activity is performed, as is the test planning activity for acceptance testing. In most cases, activities neither begin nor end precisely at the phase boundaries; rather, they overlap adjacent phases. Various methods may be used in performing an activity. For example, object-oriented design is one proven design method; structured design is another.

This document does not mandate any particular lifecycle model, and the order of activities described here is not intended to conform to any particular model. Few specific methods are mandated for required activities. These decisions are left to the project manager, who selects an appropriate lifecycle model and activity-related methods and defines them in the Project Plan. This section gives guidance on selecting an appropriate, recommended lifecycle model. For convenience, Table A-3 provides the definitions of several important terms used extensively in this section.

Table A-3. Defining a Lifecycle

Term	Definition
Software lifecycle	The period of time that begins when a software product is conceived and ends when the software is no longer available for use. A lifecycle is typically divided into lifecycle phases.
Lifecycle model	A framework on which to map activities, methods, standards, procedures, tools, products, and services (e.g., waterfall, evolutionary).
Lifecycle phase	A division of the software effort into nonoverlapping time periods. Lifecycle phases are important reference points for the project manager. Multiple activities may be performed in a lifecycle phase; an activity may span multiple phases.
Activity	A unit of work that has well-defined entry and exit criteria. Activities usually can be broken into discrete steps.
Method	A technique or approach, supported by procedures and standards, that establishes a way of performing activities and arriving at a desired result.

The lifecycle models listed below are summarized in the following subsections.

- Waterfall development lifecycle
- Incremental development lifecycle
- Evolutionary development lifecycle
- Package-based development lifecycle
- Legacy system maintenance lifecycle

A.1.2.1 Waterfall Development Lifecycle

The waterfall (single release) lifecycle is essentially a once-only, sequential-step approach. In simple terms, determine user needs; define requirements; design the system; implement the system; and test, fix, and deliver the system. Table A-4 outlines the advantages, disadvantages, and best circumstances under which to use the waterfall development lifecycle model. This lifecycle is illustrated in Figure A-1. Products and milestones for this lifecycle are summarized in Table A-5.

Table A-4. Summary of Waterfall Development Lifecycle

Advantages	Model is well studied, well understood, and well defined. Easy to model and understand. Easy to plan and monitor. Many management tools exist to support this lifecycle model.
Disadvantages	Most, if not all, requirements must be known at the start. Model does not readily accommodate requirement changes. Product is not available for initial use until the project is nearly done.
Most appropriate when . . .	Project is similar to one done successfully before. Requirements are stable and well understood. Design and technology are proven and mature. Total project duration is relatively short (less than one year). Customer does not need any interim releases.

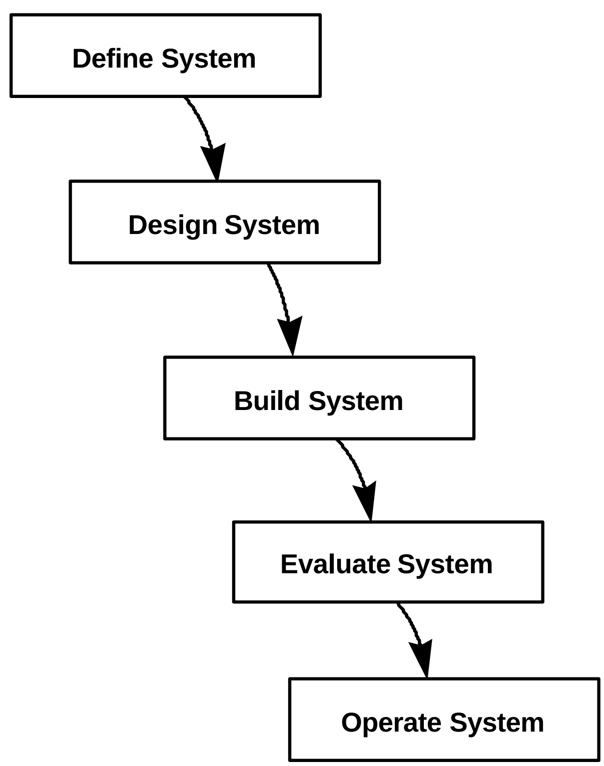

Figure A-1. Waterfall Development Lifecycle

Table A-5. Products and Milestones for the Waterfall Development Lifecycle

Lifecycle Phase	Major Products	Milestones
Initiate Project	Needs Statement Feasibility Study Benefit/Cost Analysis Risk Analysis System Decision Paper Project Plan	Initiate Project Phase Review
Define System	Functional Requirements Document Data Requirements Document Requirements traceability matrix	Define System Phase Review
Design System	System/Subsystem Specifications Database Specifications Program Specifications Updated requirements traceability matrix Acceptance Test Plan Preliminary User's Manual Prior phase products updated	Design System Phase Review
Build System	Unit-level design Implemented, tested software Database Validation, Verification, and Test Plan and procedures Updated requirements traceability matrix Operations Manual Maintenance Manual Prior phase products updated	Build System Phase Review
Evaluate System	Test Results and Evaluation Report Installation and Conversion Plan Final User's Manual Updated requirements traceability matrix Prior phase products updated	Evaluate System Phase Review
Operate System	System problem reports	Project Closeout

A.1.2.2 Incremental Development Lifecycle

For the incremental (multiple small releases) lifecycle, determine user's needs, define system requirements, and perform the rest of the development in a sequence of small releases. The first small release incorporates part of the planned capabilities, the next small release adds more capabilities, and so on, until the system is complete. Table A-6 lists the advantages, disadvantages, and the best circumstances under which to use the incremental development lifecycle model. This lifecycle is illustrated in Figure A-2. Major products and milestones for this lifecycle are summarized in Table A-7.

Table A-6. Summary of Incremental Development Lifecycle

Advantages	Reduces risks of schedule slips, requirements changes, and acceptance problems. Increases manageability. Facilitates feeding back changes in subsequent small releases. May be delivered before the final version is done, allowing end users to identify needed changes. Breaks up development for long lead-time projects. Allows users to validate the product as it is developed. Allows software team to defer development of less understood requirements to later releases after issues have been resolved. Allows for early operational training on interim versions of the product. Allows for validation of operational procedures early. Includes well-defined checkpoints with users via reviews.
Disadvantages	Like the waterfall lifecycle, most, if not all, requirements must be known at the start. Sensitive to how specific releases are selected. Places products (particularly requirements) under configuration control early in lifecycle, thereby requiring formal change control procedures that may increase overhead, particularly if requirements are unstable.
Most appropriate when . . .	Project is similar to one done successfully before. Most of the requirements are stable and well understood, but some may not yet be identified. The design and technology are proven and mature. Total project duration is greater than one year or customer needs interim release(s).

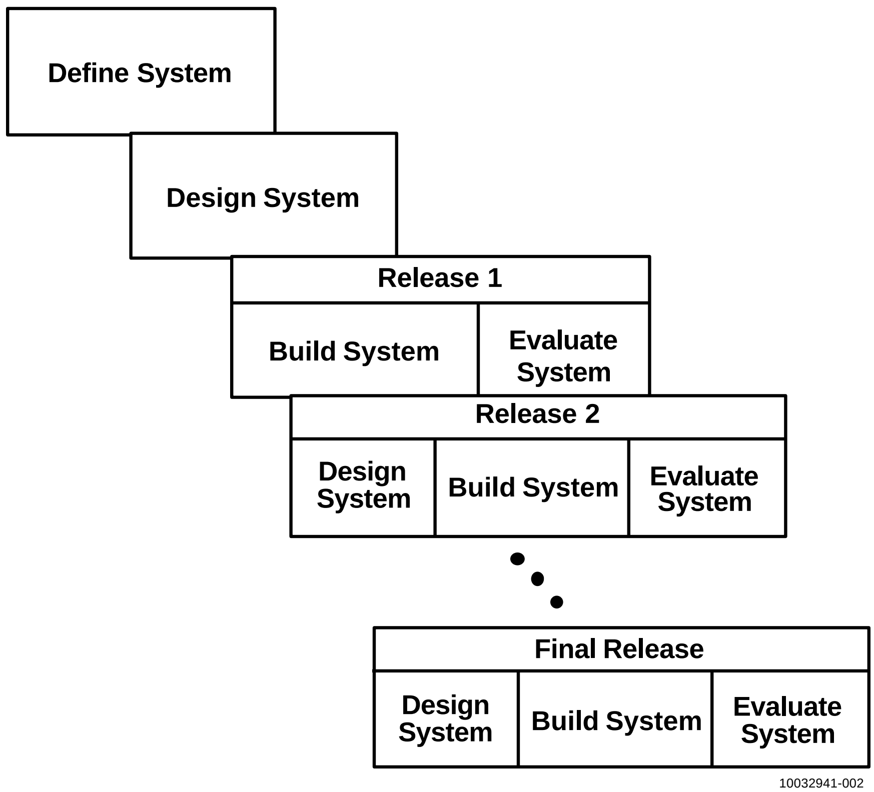

Figure A-2. Incremental Development Lifecycle

Table A-7. Products and Milestones for the Incremental Development Lifecycle

Lifecycle Phase	Major Products	Milestones
Initiate Project	Needs Statement Feasibility Study Benefit/Cost Analysis Risk Analysis System Decision Paper Project Plan	Initiate Project Phase Review
Define System	Functional and data requirements documents Requirements traceability matrix	Define System Phase Review
Design System	System/subsystem specifications through next small release Database specifications through next small release Program specifications Updated requirements traceability matrix Acceptance Test Plan Preliminary User's Manual Prior phase products updated	Design System Phase Review
Build System	Unit-level design Implemented, integrated, and tested software Database Validation, Verification, and Test Plan and procedures Updated requirements traceability matrix Operations Manual Maintenance Manual Prior phase products updated	Build System Phase Review
Evaluate System	Test Results and Evaluation Report Installation and Conversion Plan Final User's Manual Updated requirements traceability matrix Prior phase products updated	Evaluate System Phase Review
Operate System	System problem reports	Project Closeout

Return to Title Page

A.1.2.3 Evolutionary Development Lifecycle

Like the incremental development lifecycle, the evolutionary lifecycle also develops a system in small releases; but it differs from the incremental lifecycle in acknowledging that user needs are not fully understood and not all requirements can be defined at the start. In the evolutionary approach, user needs and system requirements are partially defined at the start, then are refined in each succeeding small release. The system evolves as the understanding of user needs and the resolution of issues occurs. Prototyping is especially useful in this lifecycle model. Table A-8 points out the advantages, disadvantages, and the best circumstances in which to use the evolutionary lifecycle development model. This lifecycle is illustrated in Figure A-3. Major products and milestones for this lifecycle are summarized in Table A-9.

Table A-8. Summary of Evolutionary Development Lifecycle

Advantages	Not all requirements need be known at the start. Like the incremental lifecycle, interim small releases of the product facilitate feeding back changes for subsequent small releases. Users are actively involved in definition and evaluation of the system. Prototyping techniques enable developers to demonstrate functionality to users with minimal effort. Even if time or money runs out, some amount of operational capability is available.
Disadvantages	Because not all requirements are well understood at the start, the total effort involved in the project is difficult to estimate early; therefore, expect accurate estimates only for the next cycle, not for the entire development effort. Less information is available for managing (progress is difficult to measure). There is risk of never-ending evolution (e.g., continual “gold plating”). May be difficult to manage when cost ceilings or fixed delivery dates are specified. Will not be successful without user involvement.
Most appropriate when . . .	Requirements or design are not well defined, not well understood, or likely to undergo significant changes. New or unproven technologies are being introduced. System capabilities can be demonstrated for evaluation by users. Diverse user groups exist with potentially conflicting needs.

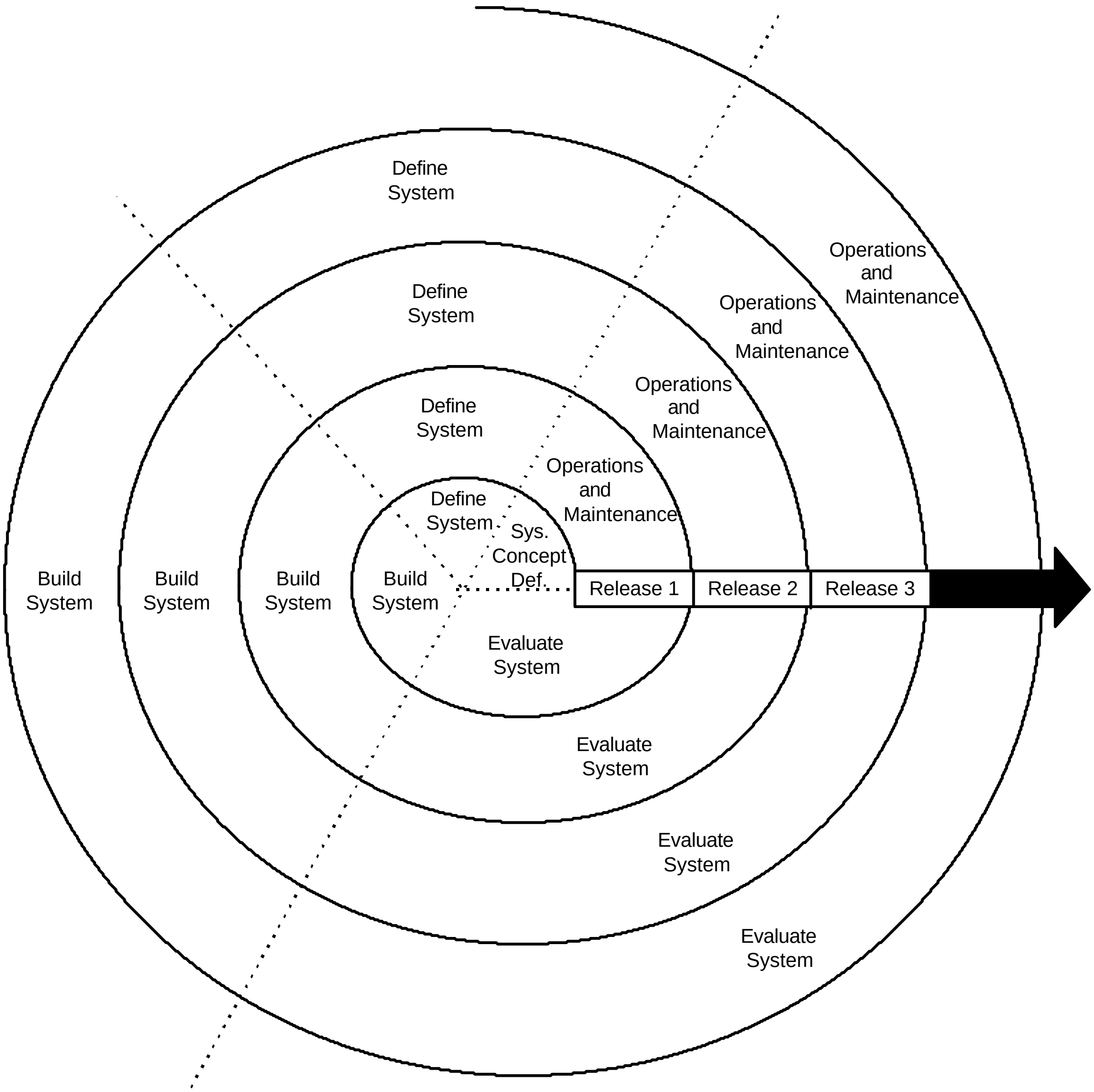

Figure A-3. Evolutionary Development Lifecycle

Return to Title Page

Table A-9. Products and Milestones for the Evolutionary Development Lifecycle

Lifecycle Phase	Major Products	Milestones
Initiate Project	Initial Project Plan, to be updated in later phases	Initiate Project Phase Review
Define System and Design System	Preliminary requirements document, including functional and database requirements Architectural design document containing the infrastructure plus the architecture of each release as it evolves Requirements traceability matrix	Combined Define System and Design System Phase Review
Build System	Prior phase products updated Software product baseline combining new, reused, and off-the-shelf products Updated requirements traceability matrix Draft User's Manual Integrated, tested software Acceptance test plan Prior phase products updated	Integration testing after all small releases have been completed
Evaluate System	Test Results and Evaluation Report Installation and Conversion Plan Final User's Manual Updated requirements traceability matrix Prior phase products updated	Evaluate System Phase Review
Operate System	System problem reports	Project Closeout

A.1.2.4 Package-Based Development Lifecycle

The package-based development lifecycle for system development is based on the use of commercial off-the-shelf (COTS) and Government off-the-shelf (GOTS) products and reusable packages. Table A-10 summarizes the advantages and disadvantages and indicates when the package-based development lifecycle model is most appropriately used. This lifecycle is illustrated in Figure A-4. Major products and milestones for this lifecycle are summarized in Table A-11.

Table A-10. Summary of Package-Based Development Lifecycle

Advantages	Lower cost than developing equivalent functionality from scratch Cycle time often lower than developing equivalent functionality from scratch More confidence in quality of the end product (because quality of reusable products is already known)
Disadvantages	May result in compromises between desired functionality and functionality provided by reusable products Maintainability may be more of a challenge because source of reusable products may not be the same organization (for example, requires third party to make changes, raises CM issues when vendor releases updated versions)
Most appropriate when . . .	A significant portion of the functionality of a system can be provided by reusable products

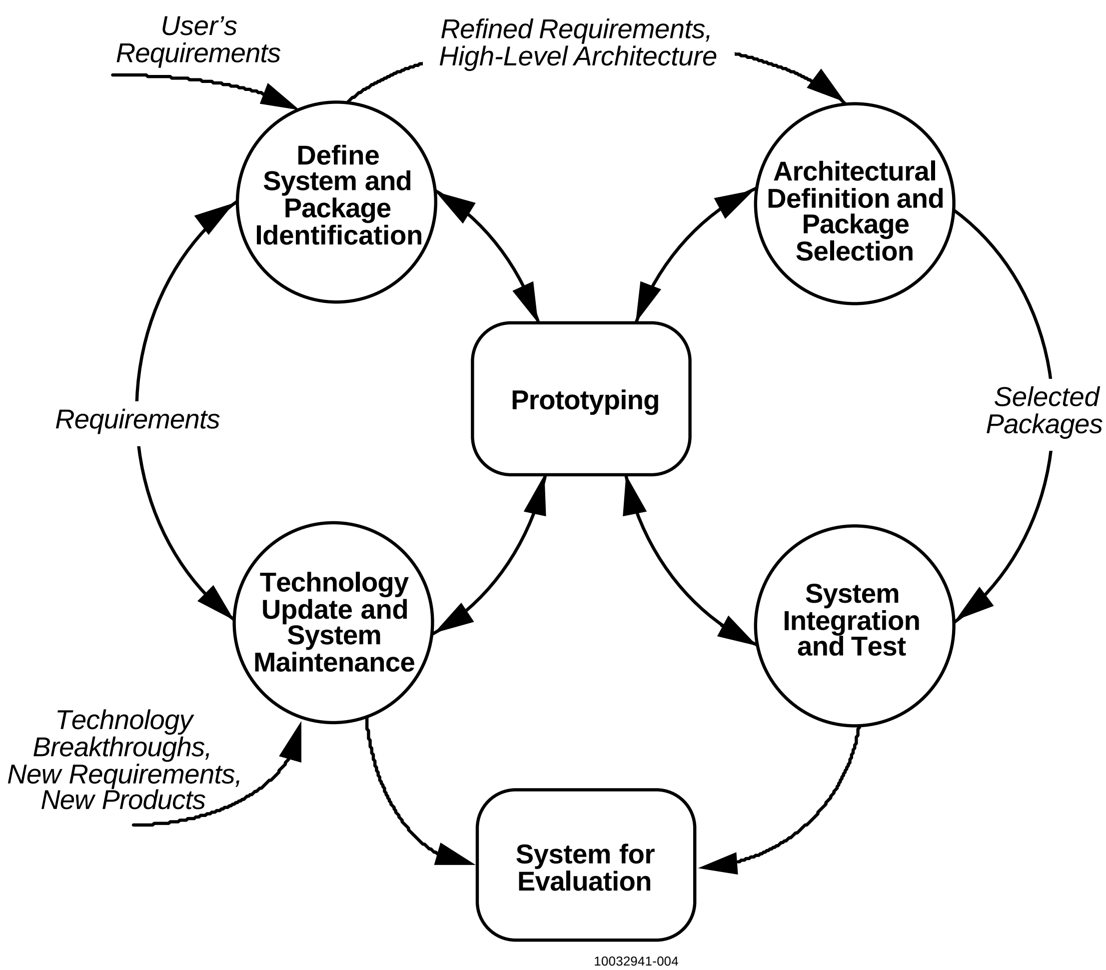

Figure A-4. Package-Based Development Lifecycle

Table A-11. Major Products and Milestones for the Package-Based Development Lifecycle

Lifecycle Phase	Major Products	Milestones
Define System and Package Identification	Requirements High-level architecture Candidate COTS packages	Define System Phase Review
Architectural Definition and Package Selection	Modified requirements System architecture Final packages	Design System Phase Review
System Integration and Test	System for evaluation	User demonstrations Build System Phase Review
Technology Update and System Maintenance	Enhanced system for evaluation	User demonstrations Build System Phase Review

A.1.2.5 Legacy System Maintenance Lifecycle

The legacy system maintenance release lifecycle is used to apply fixes or minor enhancements to an operational system. Use a waterfall or incremental lifecycle for major enhancements. Selected and sometimes abbreviated activities performed in the software development lifecycles are also performed during maintenance. The legacy system maintenance lifecycle is similar to the waterfall lifecycle; the primary difference is that in the legacy model the architectural design has already been established. This lifecycle is illustrated in Figure A-5. Major products and milestones for this lifecycle are summarized in Table A-12.

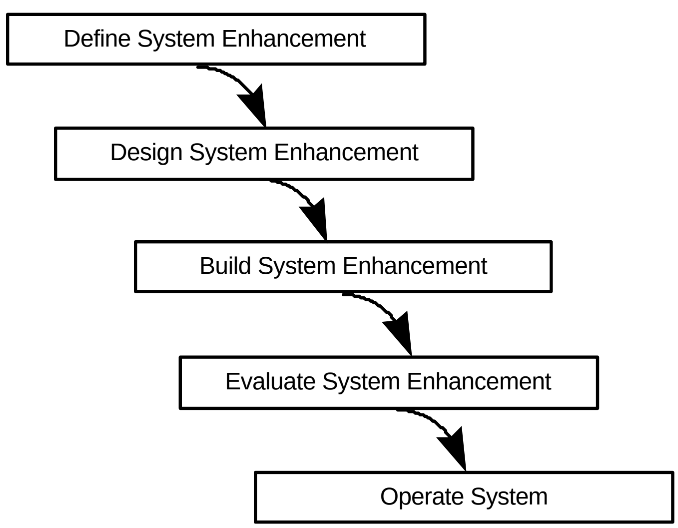

Figure A-5. Legacy System Maintenance Lifecycle

Table A-12. Products and Milestones for the Legacy System Maintenance Lifecycle

Lifecycle Phase	Major Products	Milestones
Initiate Project	Release Contents Agreement	Release Contents Review
Define System	Functional Requirements Document (updated) Data Requirements Document (updated) Updated requirements traceability matrix	Define System Phase Review
Design System	Design specifications (system/subsystem, database, program) (updated) Updated requirements traceability matrix	Design System Phase Review
Build System	Unit-level design Implemented, tested software Updated requirements traceability matrix Acceptance Test Plan Draft User's Manual updates	Build System Phase Review
Evaluate System	Acceptance tested software Test Results and Evaluation Report Final User's Manual updates Updated requirements traceability matrix	Evaluate System Phase Review
Operate System	Post-implementation review report	Release Sign-off

A.1.3 Organize Work into Measurable and Manageable Packages

The project manager determines the lifecycle model to be used, the products to be developed, and the activities and methods for developing the products. The project manager organizes the products and activities into work packages, logically organizes the work packages into a work breakdown structure, and associates cost accounts to this structure. The SDM provides guidance for the project manager on which activities should be performed to develop and maintain IT systems and on which methods should be used to meet project objectives.

Defining Measurable Units of Work

For each product to be developed and delivered, describe the specific work that must be accomplished. Document the work as a work package, including a definite schedule and time-phased budget. Define all related work packages necessary to complete the project objectives, including such items as tradeoff studies, engineering analyses, and documentation. Structure the work packages logically within the framework of a work breakdown structure (WBS) and associated cost accounts to promote effective organizing, planning, monitoring, controlling, and reporting.

Create the WBS

A WBS is a tree-structured, product-oriented, graphic representation of the hardware, software, data, and services that result from efforts to completely define the project's requirements. The WBS provides a comprehensive and consistent framework for correlating services, schedules, staffing, and costs associated with organizing, planning, monitoring, controlling, and reporting the status of each work item in the project.

At the highest levels, the WBS consists of elements that reflect the manner in which the project work is to be performed. The top-level WBS has several important uses, as follows:

- A way to make sure required work and work products have been accounted for and planned
- The basis for organizing the project team and assigning responsibilities for completing specific work
- A communication tool to show the project at a glance
- A scorecard during the project to evaluate progress against plan
- A uniform structure for collecting resource expenditures in a consistent manner across projects, thus supporting comparisons of estimated costs for new projects with actual expenditures on similar past projects

The WBS is extended to successively lower levels until the lowest elements represent the following:

- Manageable units of work that can be scheduled and budgeted
- Progress towards completion that can be measured
- Actual costs that can be accumulated.

Figure A-6 shows a sample WBS for a HUD project.

Define Cost Accounts

Cost accounts are the lowest level elements in each branch of a WBS and represent a set of management collection and control points for work. Performance measurement is applied at the cost account level. After the project manager establishes the WBS, a cost account is assigned to each work element in the WBS framework. For a small, low-risk development project, the project manager may assign multiple WBS elements to a single cost account. The project manager uses the cost accounts as the cost accumulation centers for all effort expended to manage the project. See Appendix F for detailed information on WBS structure and the associated dictionary.

A. Project Initiation/Planning

B. Requirements Definition

C. System Design

D. Software Acquisition

E. Hardware/Infrastructure Acquisition

F. New Development/Perfective Maintenance

G. System Integration and Testing

H. Installation and Deployment

I. System Operations

J. Corrective and Adaptive Maintenance

Figure A-6. Sample Work Breakdown Structure for a HUD Project

A.1.4 Assign Project Team

The project manager identifies the personnel that compose the project team. The project team is assigned with clearly stated responsibility for each deliverable and critical function of the project. Also, clear lines of communication are established with HUD IT staff and the project sponsor and users.

Identify team members and match their skills to specific work required during the project lifecycle. The project team, composed of personnel crossing internal and external organizations and functional areas, has clearly defined roles and responsibilities for project development. Work assignments should be made, and each should be clearly defined, documented, and communicated to all team members.

A.1.5 Describe Roles and Responsibilities

Throughout the project lifecycle phases, key personnel are required to perform various function-related activities. A determination is made regarding the organizations that will participate in the project and the staff resources to be provided in terms of numbers, specific skills and experience, and duration of participation throughout the project lifecycle. Use the SDM to determine the responsibilities of each functional area in the lifecycle phases. The following is a list of functional areas required for a project:

- Project Sponsor
- Project Manager
- Project Development Team (User and Developer)
- CSG
- ADP Security
- LAN Support
- Quality Assurance

Return to Title Page

- Configuration Management
- Project and Enterprise Data Administrator
- Project Database Administrator
- Enterprise Lotus Sever Administrator
- Enterprise Network Administrator

Each functional area, shown in the above list, may not be required by all projects; and some functional areas may be required only during certain phases of the project. The project manager coordinates with the functional organizations to obtain the needed resources during periods when work is defined for that specific area. Evaluate the needs for functional area support throughout the project lifecycle to correspond directly to the changing needs of the project.

A.1.6 Determine Resource Requirements

The project manager prepares estimates of the resources necessary to complete the project. As part of the estimation activity, consider all areas of the project that might expend resources, including staffing requirements, travel and training requirements, user area and development-related expenditures, software and hardware system support, and additional equipment and office space. Use measurements and cost accounting data from previous projects of similar scope to support estimates for this project.

Estimate Resources for Early and Later Phases

Estimates for the early lifecycle phase activities should be reasonably accurate. Resource estimates for later project activities may be high level and require refinement and re-estimation as the project definition is understood better and additional resource requirements are identified.

Evaluate and Select Automated Tool and Technique Capabilities

Identify the characteristics of automated tools and techniques that would be most beneficial to the project development effort. Develop a criteria for determining which automated tool or technique characteristics would be most beneficial to the development effort.

Automated tools and techniques that have been used successfully to support HUD mainframe project requirements are LINC, COMPOSER, and IEF. For LAN-based systems, consider using Powerbuilder to aid in the design and prototyping of the user interface. Evaluate the automated tools and techniques against project-driven selection criteria and determine if an automated tool or technique would be beneficial. Prepare a list of capabilities of the automated tools and techniques under consideration for use by the project. Document the results of the automated tool and technique selection process in the Project Plan.

A.1.7 Identify Special Planning Needs

Identify project considerations that warrant special attention. The nature of these special considerations varies based on the requirements of the project, its size and duration, the complexity of the work, the schedule and budget considerations, and other factors.

Use the initial work analysis to identify any of the project's special needs or characteristics of the project, such as needs for new technology development, external schedule dependencies, or special skills or training required to perform the work. Based on identified needs, establish special (risk) plans to meet these needs.

A.1.8 Establish Process for Project Reviews

As part of project planning, the project manager must include document reviews in the project schedule. Such reviews ensure that the system ultimately built fulfills its intended purpose and functionality and that key issues are identified and addressed appropriately early in the lifecycle, to avoid expensive rework in later activities. These document reviews provide feedback to the project team and assist the user organization in supporting required system approvals.

Review Level

The project manager identifies the required document product reviews and determines the appropriate level of reviews needed for the project. This determination is based on guidelines established by HUD IT. Select the organizations and individuals who are to participate in project reviews and related activities early during project planning.

Review Procedures

The project manager has the lead role in scheduling and conducting the reviews, which should conform, at a minimum, to the following activities:

- Prepare documentation for review. Prepare the technical and document deliverables from the lifecycle phase, according to HUD documentation standards.
- Schedule review. Select the time, date, and place of the review session. Ensure that the appropriate facilities are reserved well in advance of the scheduled review.
- Submit documentation for review. Submit all technical and document deliverables to the reviewers at least 10 days before the scheduled review date.
- Attend scheduled review sessions. Project personnel and reviewers attend the scheduled review sessions, and the appropriate project personnel present the deliverables and answer all questions posed by the reviewers.
- Follow up on review recommendations. Record recommendations that result from the review, and document the course of action pursued by the project, based on the recommendations.

For documents being reviewed, reach consensus on one of the following decisions at the end of the review session:

- The document is correct and complete, as is, without any further changes.
- Changes that need to be made are minor and do not require a further review. In this case, the updates should be made and change pages should be distributed by an agreed-upon date.
- Changes will have a major impact on the document. A second review must be scheduled. The changes must be incorporated and the resulting change pages distributed at least 10 working days before the second review.

A.1.9 Establish Process for Document Approvals

Obtaining approval of document products is not the goal of the lifecycle process but is a key means to the desired end: developing a system that meets the user's needs. Document approvals are obtained during all activities, from project initiation, to the operation phase, in order to establish project baselines and allow project stakeholders to agree on the project's direction. The project manager submits documents to the appropriate review board for approval. The following are examples of review boards:

Technology Investment Board (TIB): Responsible for ensuring that Departmental resources are directed to the highest IRM priorities of the Department and that these limited resources are efficiently utilized. Progress toward achieving key events and target dates for the project are tracked and reviewed by the TIB.

Change Control Board (CCB): Serves as the decision-making body for the program area projects. For each project, the project sponsor establishes and heads the CCB. The project sponsor appoints board members from the following organizations: program officials; system users; external stakeholders; and IT representatives. The CCB is the control mechanism for the program office that has requested the need for which the project has been initiated.

The project's CCB evaluates the scope, applicability, and effect of requested requirement changes (RC). The CCB focuses on items that could affect cost, schedules, or compliance with technical requirements. It acts on any requested RC to the system and provides change approval or disapproval based on defined strategic initiatives, program business objectives, and budgetary parameters. In addition, the project CCB meets to discuss impacts, especially schedule and cost impacts, of proposed changes from other program areas or organizations. The project's CCB has the authority to establish project baselines, initiate or change software, accept testing results, and approve the release of software into production.

Quality Assurance (QA): Responsible for coordinating and implementing software quality assurance for IT projects. QA verifies that plans, standards, and procedures are in place and can be used to review and audit the project. QA evaluates the software development and maintenance activities to verify compliance. QA also reviews the results of QA activities with IT and project personnel.

After approval, all documentation is placed under the configuration management change control process. This change control process is observed whenever any products of the project's baseline are revised. Change control activities include verifying changes made to the products, assigning a new version number to the revision, updating a log book with the change information, updating the central library with the new version, distributing copies of the new version, and archiving the old version. This process of change control over the lifecycle products continues throughout each project phase.

Approval of Documents

Distribute documents, with a management summary, to the appropriate review board for approval. Notify the review board of the scheduled approval decision date. Include a sign-off sheet for review board members to indicate approval. As part of the approval process, the review board assesses project status, reaches a consensus on the system's progress, and determines if any corrective action is needed. Assume approval from the review board if it fails to respond by the scheduled approval date or submits a "no comments" response in the sign-off sheet.

Management Summary

Prepare a management summary that highlights the modifications made to the document. Include in the summary a brief description of any conclusions that may be drawn and their potential impact on the project.

A.2 CONTROL THE PROJECT

After project planning is complete, project management becomes the iterative process of monitoring and controlling the project. The purpose of project monitoring and control is to ensure adequate, objective visibility into project performance and status so that effective and timely corrective action can be taken and objective assessments can be reported to the appropriate reporting levels.

To control a project effectively, the project manager must institute a performance measurement discipline to monitor progress regularly within each lifecycle phase and must train team leaders in the exercise of the discipline. In its most basic form, performance measurement is the comparison of a predetermined schedule for accomplishing a quantity of work with the technical performance achieved and the actual costs incurred at a specific point in time. To make this comparison, measure the schedule, performance, and actual costs in the same units; dollars is the industry-accepted standard.

During the planning phase of a project, translate the project requirements into measurable units of work. Schedule each unit of work; establish intermediate events (milestones) to measure progress toward accomplishing each unit; and assign a dollar value, representing the estimated effort required to accomplish the milestone. The cumulative value of these milestones is called the budgeted cost of work scheduled (BCWS). The cumulative value of all milestones achieved by a specific time is called the budgeted cost of work performed (BCWP), sometimes referred to as earned value. The actual cost expended in achieving the milestones, extracted from the HUD accounting system, is called the actual cost of work performed (ACWP).

Comparing BCWP and BCWS provides a measure of schedule variance (SV) showing whether work is accomplished at the rate planned. Comparing BCWP and ACWP provides a measure of cost variance (CV) showing whether work is accomplished at the rate at which costs are incurred. Comparing BCWS and ACWP provides a measure of budget variance (BV) showing whether actual costs are incurred at the rate they were planned. Table A-13 summarizes these basic concepts.

Apply the basic elements of performance measurement (BCWS, BCWP, and ACWP) to specific, defined units of work (cost accounts). Together with BCWS, BCWP, and ACWP for other units of work, create a summary to successively higher levels until the project level is reached. Periodically during the performance of a project, estimate the cost of completing all remaining units of work based on performance to date.

The project manager must be proactive in determining project status and in taking responsive corrective action when required. Do not wait until the end of a phase to document and identify project impacts that may be correctable within the phase. Isolate and address the impacts within the phase with periodic status assessment.

Table A-13. Summary of Basic Performance Measurement Definitions

Generic Element of Performance Measurement	Equivalent Term	Acronym
Work planned	Budgeted cost of work scheduled	BCWS
Work accomplished; earned value	Budgeted cost of work performed	BCWP
Cost of work accomplished	Actual cost of work performed	ACWP
Schedule variance	BCWP minus BCWS	SV
Cost variance	BCWP minus ACWP	CV
Budget variance	BCWS minus ACWP	BV

The following are activities associated with monitoring and controlling the project:

A.2.1 Collect, review, and post project data.

A.2.2 Calculate variances.

A.2.3 Evaluate project status.

A.2.4 Resolve variances and address problems.

A.2.5 Report project status.

A.2.6 Manage change and issues.

A.2.7 Manage risk.

A.2.8 Manage configuration.

A.2.9 Manage quality.

A.2.10 Measure productivity and quality.

A.2.11 Keep the Project Plan up to date.

A.2.1 Collect, Review, and Post Project Data

Collect Project Data

Collect time accounting data on the effort expended on project activities. On large projects with multiple levels of management, gathering all required information starts with the project leaders or team leaders. On small projects, the project manager typically performs this activity alone. Collect cost data on other project expenditures, such as software; equipment; and administrative, training, and travel expenses.

Review Project Data

Ensure there is accurate information to analyze and evaluate the project status and to initiate corrective action. Review expenses to verify that they can be charged to the project and are not excessive.

Post Actual Data

The project manager should use an automated tool to enter and analyze the results of actual and planned data. Periodically, update the baselined Project Plan to reflect actual costs against scheduled costs for work in the project. This allows each individual WBS work element (or cost account) to be tracked and the work to be summarized by period (e.g., weekly, monthly, by phase, or to date). For each WBS work element (or cost account), perform the following:

- Enter the data on actual effort worked.
- Designate what portion of the work is complete.
- Enter any other expenses incurred against it.

A.2.2 Calculate Variances

The project manager should calculate schedule, budget, and cost variances to assess the status of each work package in each cost account. Instruct team leaders to indicate the work completed for the recent period by using the criteria established during the planning process. Consider the work completed only if it was inspected and certified as compliant with the planned completion criteria.

Use the variance parameters for insight into the plan's effectiveness and as indicators for corrective action. Calculate the variances as a function of period (e.g., month) to ascertain trends; calculate the variances cumulatively to see where the project stands for its current timeline and schedule. Update the Project Plan and budget using the results.

Roll up this detailed information through all levels of the WBS to determine the potential impact on the Project Plan for schedule, cost, and budget. The length and complexity of this process depend on the size, organization, and complexity of the project. On large projects, the roll-up process may go through multiple levels of managers and team leaders because the project may have intermediate levels of schedules and budget. Except for very small projects, if possible, use an automated tool to streamline the process and reduce the chances for error.

A.2.3 Evaluate Project Status

Assess whether the project will be completed on schedule and within budget. Do not rely on intuition or subjective analysis. Instead, evaluate the project status objectively by using factual data. The information available for evaluation includes the following:

- Project schedule in the Project Plan
- Project budget
- All work (cost account) schedules
- Work status reports
- Previous project status reports
- Reports and logs

If the project status indicates a problem and if the solutions are within the project manager's authority, select the best alternative and make the senior managers, project sponsor, and user aware of the decision. If the problem is significant and the potential solutions are beyond the project manager's authority, present them to the project sponsor, user, and senior management for a decision. This does not imply that the project manager is not involved in the decision. The project manager needs to assist senior management and the user in making the right decision.

Review each WBS work element (cost account) for variances and causes of the variances. See Table A-14 for typical causes. If a problem is found, examine it with the team leader responsible for that work and determine resolution. Some situations exist where no change is necessary. Some minor deviations from start and completion dates can be allowed as long as they are justified. Some positive variances are acceptable if they are offset by negative variances. However, because of the potential adverse impact, do not be satisfied by a cursory examination of minor deviations. They can add up to a big problem.

Evaluate the overall project status by analyzing the schedule, budget, and cost variance trends. Variances by themselves are neither good nor bad. They are simply data that project managers use to ask intelligent questions about project status. Examine the variance trends periodically (weekly, monthly) to determine in what direction they are heading or whether a significant change has occurred. In addition, by analyzing the variances on accomplished work, variances can be estimated on work that remains to be completed.

In addition to looking at variances and problems individually, look at them collectively to assess their impacts on the overall project. Collective impacts include the following:

- Problems are not identified quickly because team members are dealing with symptoms rather than causes.
- Problems are being solved too slowly because the solution process is too complex.
- Problems are resurfacing because initial solutions are inappropriate or are being communicated ineffectively.
- Problems or solutions are not under the control of the project team.

Table A-14. Causes of Variances

Typical Causes of Schedule Variances	Typical Causes of Cost Variances
Unexpected problem Complexity of task Poor communication and coordination Unplanned necessary work (if it impacts planned work) Unnecessary add-ons Overdesign, overanalysis Changes in requirements Staffing shortage Low productivity (for specific reason) User/sponsor-directed change in approach/priority	Unexpected problem User-directed change in approach or priority Complexity of work Unplanned necessary work Unnecessary add-ons Overdesign, overanalysis Changes in requirements Inability to proceed (for specific reason), yet requirement remains to maintain staffing level Higher skill mix required (for specific reason) Unplanned reiteration of previously completed work Unplanned other direct cost, e.g., software or hardware purchase not initially planned

A.2.4 Resolve Variances and Address Problems

If the overall assessment of variances indicates that the project is over budget or behind schedule, corrective action must be taken. Keep in mind that a corrective action to resolve one type of variance (e.g., schedule) may adversely impact another type (cost). Consider these options for corrective action:

- Change work (cost account) assignments. Redistribute some work.
- Combine previously separated activities or products.
- Fix specific problems that prevent recurring slippage.
- Implement specific workarounds.
- Upgrade or downgrade skill mix. (At the expense of budget, bring in senior people who have the experience to gain schedule; to save budget on simple work, bring in junior people.)
- Increase staff size. (Once a project is underway, however, increasing staff never helps as much as expected. Extra staff members need offices and equipment, add to management overhead, need training, and usually require user and senior management approval.)
- Work longer hours. (Explain the situation to the team and ask them to pitch in; however, over a long period of time, this will usually burn out the team.)
- Reduce the project scope; this is a major decision requiring user approval.
- Extend the completion date or increase the budget; these are major decisions requiring senior management and user approval.

Based on corrective action, calculate a projection of the staffing level required to complete the remaining work. Use the cost (productivity) measurement data from the work already performed to calculate the cost of the remaining work for the projected staffing. If it deviates significantly from the Project Plan, consider alternative corrections or revisit the Project Plan with senior management and users for replanning.

Sooner or later, no matter how well a project is managed, a project runs into trouble. Following good project control standards will at least help in recognizing trouble early and allowing appropriate action to be taken.

A.2.5 Report Project Status

The objectives of tracking and formally reporting project status are as follows:

- Provide a consistent technique for monitoring progress against plan.
- Identify problems quickly to allow maximum time for corrections.
- Provide an objective rather than subjective evaluation of status.
- Give the project sponsor, users, support organizations, senior management, and other reporting levels timely information.
- Provide a managerial evaluation rather than just raw facts.

Reviews and reporting occur throughout the project. They are conducted and written for different audiences, such as project sponsors and users, senior management, and the project team. Sometimes, senior management may require special reports, such as for project budgeting. Within the project, define and implement applicable project status reporting.

Wherever possible, synchronize the reporting needs and cycles of the user and senior management so that all requirements can be met with a single set of reports.

Determine Status Reporting Hierarchy

On large projects with multiple levels of management, define a reporting structure that coincides with the project organization. Within this structure, each manager and leader prepares appropriate status reports for their reporting level. Coordinate among the reporting levels to determine reporting media (hardcopy, electronic, etc.). Figure A-7 illustrates the reporting hierarchy.

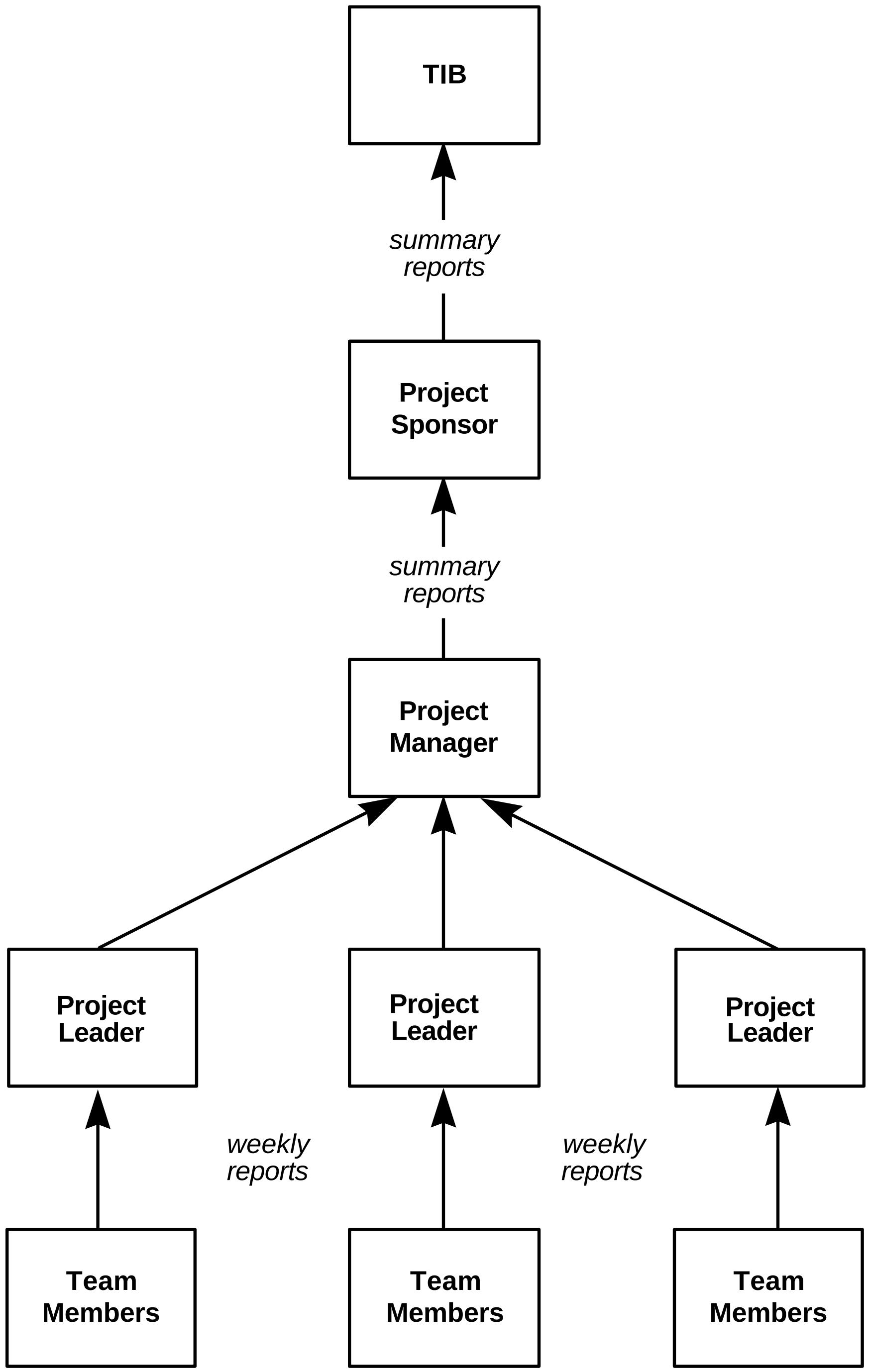

Figure A-7. Reporting Hierarchy

Summary Report Contents

The project manager prepares summary reports for senior management, the project sponsor, and others, as appropriate. Summary reports from the project manager will vary in content to satisfy the requirements of the reporting levels.

- For the project sponsor, provide schedule and issues, variances and corrective action.
- For the TIB, provide a quarterly status report of schedule and actual costs in dollars and hours.

Conduct User Review Meetings

In addition to providing a forum for reviewing project status, regular review meetings with the sponsor and other support groups (e.g., ADP Security Office) fulfill a number of objectives. They serve to keep key people and organizations informed of pending resource needs, establish working relationships, and facilitate making routine decisions. Schedule regular review meetings with the project sponsor, support organizations, and users.

A.2.6 Manage Change and Issues

The project manager is responsible for monitoring and controlling changes and issues that arise in the course of the project.

Manage Changes

Changes are those activities not originally considered within either the products to be delivered or work to be performed as defined by the project's original scope. Fill out a change report when a change is identified. Review the reports written and submitted by users and team members. Use a log to enter and track the reports. Update the report entries as the status changes. Changes can require additional time and budget. Examples of changes include the following:

- Development of processes not previously identified
- Participation in activities not previously identified, such as making presentations or demos to external organizations not identified with the project

Manage Issues

Every project has unexpected issues that arise to hinder progress. These issues are handled by action items to the appropriate functional organization. The project manager needs to ensure that these issues are identified and resolved quickly so that the project can proceed according to plan. To do this, follow these guidelines:

- Identify issues that are detrimental to meeting the objectives of the Project Plan.
- Document the issues in a log, associate responsibilities by action items, and track the status of issues.
- Update this log as old issues are resolved and new issues are identified.
- Include issue resolution progress as part of your normal project status reporting.
- Regularly review the log with the project sponsor, support organizations, and users.

A.2.7 Manage Risk

Risks are issues whose magnitude of scope may cause schedule and budget variances that cancel the project if risk mitigation is not identified and applied to resolve the risk. Identify risks and their mitigation at the beginning of, and throughout, the software development lifecycle.

Risk management requires that the project manager identify and analyze risks and plan to mitigate risks throughout the project software lifecycle. Periodically, identify special project needs that could represent areas of risk. Examples of risk items are the following:

- Any needs for new technology development
- Any schedule dependencies upon events outside of a project manager's control, such as undelivered hardware or insufficient computer test time and equipment
- Any special skills required to perform the work

Analyze each risk item to estimate the probability of occurrence and the potential cost, schedule, or technical impact. For each risk item:

- Establish bounds beyond which the risk cannot be tolerated, and determine the best approach to mitigate the risk.
- Determine the required resources, schedules, monitoring techniques, and associated cost of each action of the potential mitigations.
- Identify early how and when progress will be reported to highlight any problems associated with risk.
- Close out the risk when it no longer exists.

For medium and large projects, document the risk assessment and mitigation plan in a risk management plan.

A.2.8 Manage Configuration

The objectives of managing the project configuration are to establish baselines and to ensure that the changes proposed to the baseline are necessary and appropriate, that the integrity of the system is maintained, and that a record of changes to the system is kept. The project configuration is made up of all components required to develop and operate the system successfully. These components include hardware, software, facilities, work products, and documentation. Managing changes to the system is accomplished by managing changes to these components. The project CCB is established as the control mechanism for baselining system products. In addition, the project CCB reviews for approval (and implementation) any changes to the baseline. Create a CM Plan (CMP) for describing CM activities. If an existing project CMP is available and if the CM activities in the plan are applicable, use it for the project. Otherwise, document the CM activities for the project in a CMP during the Define System phase. Receive approval for the CMP from the project CCB.

The major activities of CM are as follows:

- Establish CCB and CCB interfaces for the project.
- Establish a central project repository (library).
- Determine the project baselines.

Return to Title Page

- Identify the configurable items.
- Define and implement configuration change controls.
- Monitor and report configuration status.
- Manage changes to project documentation.
- Audit configuration changes.

A.2.9 Manage Quality

The QA function needs to be part of every project, although the formality and size of the function will vary from project to project. In general, as the user's requirements become more rigid, as the project team becomes larger, or as the complexity of the system being delivered increases, the need for a more formal, larger QA function increases accordingly.

Every project must have a person who defines and supports the quality process within the project. On a small project, this is usually the project manager. For larger projects, a full- or part-time quality officer should be appointed to assume the quality management responsibilities. Depending on the amount of work involved, the quality officer may have a team of people to support the QA function.

Coordinate with the QA organization for assistance to develop a QA Plan that describes what quality is for the project and how it will be achieved in a quality program. Tailor the IT QA Plan, if applicable, to develop a project QA Plan; or, if an existing branch or division QA Plan is available and the QA activities in the plan are applicable, use it for the project. Otherwise, document the QA program for the project in a QA Plan during the Define System phase. The actual format, size, and detail of the QA Plan will vary according to the project size, complexity, and user requirements. Interface with the QA organization for approval of the project QA Plan. For the quality program:

- Identify the QA organization and roles.
- Describe the standards, procedures, training, and tools.
- Identify the processes and indicators that will be used to measure quality.
- Receive approval for the QA Plan from the QA organization.
- Monitor process effectiveness.

A.2.10 Measure Productivity and Quality

Establish a process to capture measurements that characterize productivity and quality. Use the results to estimate future project schedules and to verify effectiveness of pilot programs.

Typical measurements to determine productivity are as follows:

- Lines of code
- Function points
- Screens
- Scripts

Return to Title Page

- STARS
- Combination of above

Use an appropriate unit of measurement for effort, such as staff hour or staff month, to calculate the productivity rate.

As an example for quality, the measurement unit may be based on errors detected or mean time between failure (MTBF). Use meaningful units to correlate a quality rate to the product; for example, STARS per product size.

Document the productivity and quality measurements of the effort for use by other development and maintenance software projects within the organization. If the measurement data are updated and maintained in a project system decision paper, as defined in the SDM, then take ownership of the paper for application to other similar projects. Otherwise, prepare a software project history in the Project Plan that characterizes the product generated, including measurements, and processes used to accomplish the work.

A.2.11 Keep the Project Plan Up to Date

An effectively maintained Project Plan documents the current strategy for the software effort. Because it provides a uniform characterization of the project, the Project Plan can be invaluable if changes occur in team leadership.

Account for all revisions to the project in the Project Plan. Most projects undergo schedule and budget revisions across the life of the project. Be prepared to update the Project Plan at the end of each phase. Situations that can cause a revision to the Project Plan include the following:

- A change in staff size, roles and responsibilities, skills, availability, or productivity
- Significant variances in schedule or budget
- A request to alter a project milestone or completion date
- A change in project assumptions or constraints
- Refinement and a better understanding of later phases

In general, if anything occurs that changes the project team, duration, scope, or estimates, the Project Plan is subject to change.

This page intentionally left blank.

Return to Title Page

Appendix B. Configuration Management Guidelines

This page intentionally left blank.

APPENDIX B. CONFIGURATION MANAGEMENT GUIDELINES

Configuration management (CM) is the ongoing process of identifying and managing changes to deliverables and other work products as they evolve through system development and maintenance. The objectives of CM are to ensure that baselines are defined, the changes proposed are necessary and appropriate, the integrity of the system is maintained, and a record of changes to the system is kept to provide traceability throughout the lifecycle.

Although the basic elements of CM are the same, the implementation of CM activities varies depending on the type of system and environment requiring control. As such, the CM requirements imposed on a project and/or system must be selected to provide the optimal approach for fulfilling the goals of CM.

The elements for implementing CM are:

- Configuration management program
- Configuration management standards and procedures
- Configuration management measurements
- Training
- Implementation strategy

B.1 CONFIGURATION MANAGEMENT PROGRAM

Document the project's CM program in a Configuration Management Plan (CMP). Depending on the project's type, size and complexity, the CMP can be a subsection of the Project Plan or a standalone document. Use the CMP as a basis for performing CM and defining CM activities, responsibilities assigned to CM, and required resources (including staff, tools, and computer resources, if necessary). During CM planning activities, identify the required procedures needed to successfully implement the activities. Prepare the CMP in parallel during project planning. The CMP requires project and IT management approval.

Personnel responsible for coordinating and implementing CM for the project administer the activities defined in the CMP. Considerations for staffing CM include assigned tasks and activities, project size, and organization structure. CM personnel may be assigned as part time or as dedicated to the project. For example, the CM team is established at the information technology (IT) division level to provide CM support to division projects. For "large projects," a CM team can be established specifically at the project level.

The CM program that meets government and industry standards is defined in terms of six major sets of activities:

B.1.1 Configuration identification

B.1.2 System baselines

B.1.3 Software baselines

B.1.4 Configuration control

B.1.5 Configuration status accounting

B.1.6 Configuration audits

B.1.1 Configuration Identification

Configuration identification is the process of identifying items to be placed under configuration control. System development products include the following three sets of items:

- Configuration items (CIs), such as software, hardware, communications, and databases
- Technical documentation or baselines describing the CIs
- Management documentation describing the process used to develop or manage the development of the CIs, such as plans, standards, and procedures

A CI is an integrated set of system components and may be composed of any combination of components either developed by the project or purchased for the project (e.g., COTS). The types of system components vary depending on the system's operational environment and may include the following:

- ***Hardware components:*** Computer workstations, peripherals, servers, and routers
- ***Software components:*** Computer programs, operating systems, and support tools
- ***Database components:*** Files and records that exist apart from software that accesses the contents of a database

Within the HUD IT environment, CIs correlate to HUD IT systems. Each system and its components are documented via controlled specifications, drawings, or other technical documents. CIs are the application and associated system documents. Controlled documentation is the principal means for controlling changes to the system during development and maintenance. System documents are to be placed under configuration control upon management and user approval at the completion of a system development or maintenance phase. Each system is assigned a system ID, acronym, system name, purpose description, and other information used to identify the system. Some examples of current HUD system CI details are shown in Table B-1.

The results realized from comprehensive CM activities provide system traceability, where a given system or application release is linked with its CIs, modules, units, documentation, system changes (including problem fixes), impacts to external systems, and impacts from external systems.

Table B-1. HUD IT Systems as Clearly Defined Configuration Items

System ID	Acronym	System Name	Purpose
D53	MSCS	Merit Staffing Control System	Used to issue Merit Staffing Vacancy Announcements
A21	LAS	Loans Accounting System	Processes Section 202 ...
A67	LOCCS	Line of Credit Control System	Supports the Office ...
F47	MFIS	Multifamily Insurance System	Provides automated online ...

B.1.2 System Baselines

A baseline is a set of controlled documents describing the system at a specific point in the system lifecycle. Baselines are subject to configuration control and are maintained to reflect authorized changes. Plans, reports, standards, and procedures are part of baselines only if they define technical aspects of a system not otherwise specified in baseline documentation.

Two of the most common documentation events on system development and maintenance projects are establishing and documenting changes to baselines. Baselines establish a common point of reference for system development within the project and with the user and sponsoring organization. Baselines must be established before changes can be made.

Figure B-1 shows the points in the system development lifecycle at which system baselines are established. Formal reviews are conducted at the end of each lifecycle phase, at which time baselines are presented for management and user approval or concurrence. The review participants include assigned project personnel (management and technical personnel, support staff, and system users). Table B-2 shows the technical documents that comprise the system baselines.

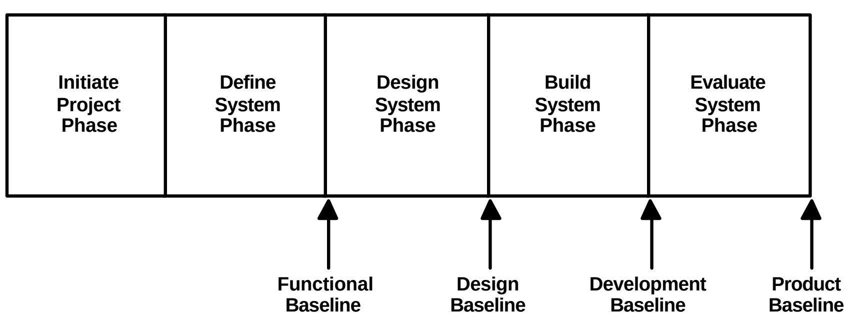

Figure B-1. System Baselines Established at Discrete Points in the Lifecycle

Table B-2. Major Technical Documents of the System Baselines

Baselines			
Functional	**Design**	**Development**	**Product**
Functional Requirements Document	System/Subsystem Specifications	Operations Manual	Test Results and Evaluation Report
Data Requirements Document	Database Specifications	User's Manual	Installation and Conversion Plan
	Program Specifications	Updated Functional and Design Baselines	Updated Functional, Design, and Development Baselines
	Test Plan		
	Updated Functional Baseline		

B.1.2.1 Functional Baseline

The functional baseline, sometimes called the requirements baseline, is the main product of the Define System phase. The main components of a functional baseline are functional requirements and data requirements documents that describe the system's functional, data, performance, interface, reliability, and security requirements. Review this baseline at the completion of the Define System phase. After making the changes needed to resolve problems found during the review, formally establish the baseline upon receipt of management approval.

B.1.2.2 Design Baseline

The design baseline culminates from activities performed during the Design System phase. Its major component is a system/subsystem specification that defines the overall system design in terms of its subsystems, the allocation of requirements to subsystems, and interfaces between subsystems and external systems. Define the user acceptance evaluation criteria component of this baseline in the VV&T Plan. The user acceptance evaluation criteria is not a separate document but is a major element of the design baseline. Review the design baseline at the completion of the Design System phase. After making the changes needed to resolve problems found during the review, formally establish the baseline upon receipt of management approval.

B.1.2.3 Development Baseline

The development baseline, generated during the Build System phase, defines the detailed structure of the system being implemented. Use stringent configuration control for the development baseline; otherwise, planning and controlling system implementation will be difficult, if not impossible. The development baseline's major component is the generation of the computer programs and the database. Other components are the training documentation and user's, operations, and maintenance documentation. Review the development baseline at the completion of the Build System phase. After implementation of any changes agreed to at the review, formally establish the development baseline upon receipt of management and user approval and concurrence.

B.1.2.4 Product Baseline

The product baseline is established during the Evaluate System phase. The product baseline's major component is the end system product as built by the developers. A second major component of this baseline includes the system user's, operations, and maintenance manuals and the system installation and conversion procedures.

The product baseline is established after successful completion of the functional configuration audit (FCA) and physical configuration audit (PCA) and the associated system products and audit results presented at the Evaluate System review. The baseline incorporates all changes needed to resolve problems detected during system acceptance testing and any discrepancies between the system and its requirements and design documentation. Following implementation of any changes agreed to at the review, formally establish the baseline upon receipt of management and user approval and concurrence.

B.1.3 Software Baselines

As the software portion of the system is implemented (e.g., coded and tested), establish controlled software libraries. Use the controlled software libraries for managing the baseline and maintaining the source code and compiled and executable code. Also use these libraries to control the migration of software as it evolves from the development system environment to the production environment. Manage the libraries using computer access controls or other automated tools, such as ENDEVOR, to protect each controlled library from unauthorized change.

Maintain three types of controlled software libraries: development, pilot, and production. The development library contains the software used through system acceptance testing. Upon successful system acceptance testing, create an intermediate pilot library for controlling software used as part of pilot testing activities. The production library contains the software used operationally in the production environment. The Change Control Board approves the migration of software from the development library to the production library.

Each software library possesses version control capabilities. This version control enables accurate locating of the latest version of a program within the software library and enables CM personnel to regenerate prior software configurations if new versions fail. The product baseline is the final system baseline. After the system becomes operational, system enhancements or problem corrections may be required. For each modified release, update the affected system documentation. A new product baseline supersedes the previous one.

B.1.4 Configuration Control

Effective configuration control depends on placing products under control at the right time and on establishing mechanisms for controlling changes to the products. Placing a product under formal configuration control too early can create a heavy (and unnecessary) administrative burden in tracking changes before the item stabilizes. Placing it under formal control too late reduces management visibility in the development process, permits uncontrolled changes, may disrupt other parts of the project if different versions of the system are used, and can effectively circumvent quality and configuration controls.

Place documentation deliverables under configuration control when they have undergone a lifecycle review. Provide documentation deliverables to management and users for approval or concurrence via the CCB. Place software, hardware, communications, and database components under control to start system testing.

The following five basic mechanisms are central to configuration control:

B.1.4.1 Change Control Board

B.1.4.2 Needs Statement (NS)

B.1.4.3 Requirements Change (RC)

B.1.4.4 Advanced Requirements Notice (ARN)/Evaluation of Request (EOR)

B.1.4.5 Service Ticket Action Resolution System (STARS)

The project CCB must approve all changes to management-approved and user-approved baselines. Change requests and problem reports for all controlled system products and deliverables are recorded, reviewed, approved, and tracked to closure. The following sections describe an effective approach to controlling system changes. The exact details of the implementing procedures may vary depending on organizational divisions.

B.1.4.1 Change Control Board

The CCB serves as the decision-making body for the program area projects. For each project, the project sponsor establishes and heads the CCB. The project sponsor appoints board members from the following organizations: program officials; system users; external stakeholders; and IT representatives. The CCB is the control mechanism for the program office that has requested the need for which the project has been initiated.

The project's CCB evaluates the scope, applicability, and effect of requested requirement changes (RC). The CCB focuses on items that could affect cost, schedules, or compliance with technical requirements. It acts on any requested RC to the system and provides change approval or disapproval based on defined strategic initiatives, program business objectives, and budgetary parameters. In addition, the project CCB meets to discuss impacts, especially schedule and cost impacts, of proposed changes from other program areas or organizations. The project's CCB has the authority to establish project baselines, initiate or change software, accept testing results, and approve the release of software into production.

B.1.4.2 Needs Statement

The NS is a request that describes a specific need within a functional area. An NS is required to initiate each major project or automated information system, including major system modifications and additional procurements for existing systems. Use the NS to justify the exploration of alternative solutions and obtain required TIB approval. In addition, the NS is used to initiate perfective maintenance for production systems. Perfective maintenance activities are modifications requested to improve a system in an area that does not impact functional requirements.

B.1.4.3 Requirements Change

The RC is a request to change technical requirements approved for the project. It is usually generated by system users, affects the scope of the system, and may have an impact on the project's overall cost, schedule, and technical capability. The RC must be approved by the project CCB before project resources are assigned to implement the change.

B.1.4.4 Advanced Requirements Notice/Evaluation of Request

The ARN is a work request generated by personnel within HUD who wish to initiate minor system enhancements, considered adaptive maintenance, to production systems. ARNs are normally results of user initiatives, legislative and regulatory developments, and upgrades to HUD’s system technical architecture. These adaptive maintenance changes are usually activities that have been planned for future implementation. The EOR is a documented evaluation of the project dates and costs required to complete the work defined in the ARN.

B.1.4.5 Service Ticket Action Resolution System (STARS)

STARS is the mechanism used to log requests regarding software problems and to initiate corrective maintenance for production systems. Design, logic, or coding errors usually cause software problems. The project manager determines their resolution priority. Inform the project’s CCB of all STARs (problem tracking reports) assigned to the project. Usually, only controversial STARs and those that could have major affects on costs or schedule should be subject to CCB review and approval.

B.1.5 Configuration Status Accounting

Configuration status accounting focuses on recording and monitoring changes to controlled system configurations and maintaining a controlled documentation library. Process changes to controlled software libraries and documentation using change control mechanisms. Maintain data records and prepare reports to provide an inventory of initial baseline configurations, of both software and documentation, and a record of changes made to the original configuration.

The controlled documentation library consists of all approved, baselined deliverable documents, including the initial version of each document and any subsequent document changes. Maintain CM records to provide an audit trail that identifies all changes implemented on approved baseline deliverables. In addition, maintain documentation distribution lists as part of configuration status accounting. Distribution lists include the manager or personnel responsible for the document, the signature requirements for each document, and the recipients of the document. Also maintain CM records to identify the software versions installed at HUD sites, including Headquarters and field offices.

Provide configuration status accounting reports to the appropriate project personnel to report the up-to-date status of baselined deliverables and decisions made by the CCB.

B.1.6 Configuration Audits

Perform a system FCA and PCA to confirm that the system meets its technical requirements, is accurately described in documentation, and does not include any unauthorized changes. These configuration audits are usually conducted at the completion of system integration testing and before the software is migrated to the production environment. During the FCA, evaluate the test records to determine if the system successfully fulfilled its requirements and user needs. During the PCA, assess the system technical documentation for completeness and accuracy in describing the tested system.

The FCA process includes the following activities:

- Review test and analysis results against test plans and procedures to ensure that testing is adequate and properly performed.

- Review test and analysis results to verify that actual performance of the system complies with its requirements and that sufficient test results are available to ensure that the system will perform in its operational environment.
- Review all waivers to ensure that they were approved.

The PCA process includes the following activities:

- Compare the tested system configuration with the operational system delivery to ensure that the appropriate components were tested.
- Verify that the system complies with all applicable standards.
- Review the system documentation to ensure that it accurately reflects the tested configuration.

B.2 CONFIGURATION MANAGEMENT STANDARDS AND PROCEDURES

Detailed CM standards and procedures that are needed define the specific processing steps required to effectively implement CM in different environments, namely, mainframe, LAN/PC, and existing legacy systems. The CM procedures document the methods for performing CM activities and take into account the system's environmental elements, such as distributed systems with remote sites, multiple versus single system installations, revisions, centralized software control, multiple hardware configurations, and organizational interfaces and responsibilities.

Document standards and procedures that define the specific tasks (or activities) necessary to meet the CM program requirements. Define the roles and responsibilities of all project personnel involved in a CM-related activity. In addition, use the standards and procedures to train personnel in performing CM tasks and to identify measures used to evaluate the effectiveness of the CM program.

The minimal set of CM standards and procedures address the following:

- Controlled library maintenance
- Change control mechanisms
- Processing change control mechanisms
- Configuration status account reporting
- System release
- Configuration audits

The CM standards and procedures define major activities and products required by the CM program and provide flexibility to allow the project to implement an efficient and productive program.

B.3 CONFIGURATION MANAGEMENT MEASUREMENTS

Define CM measurements to determine the status of CM activities, the effectiveness of CM processes, and the stability of controlled baseline deliverables. Target these measurements at aiding management in determining the quality of software releases after changes are implemented. Using CM measurement data and reports, project managers and staff have the capability to do the following

- Perform trend analysis on the effects of prior software changes.
- Determine the effectiveness of the overall CM program, especially the software change process.
- Classify the types of software changes to determine where maintenance resources are concentrated.
- Identify programs and module routines requiring frequent modification and rework.
- Analyze maintenance activities to evaluate the stability and predictability of programs.
- Make informed decisions about the future of systems.

This type of information will facilitate meaningful analysis of CM data and allow management to focus on areas requiring attention or corrective action.

B.4 TRAINING

Provide configuration management training to all personnel supporting the project that is tailored to the CM program. Train project personnel, including those assigned responsibility for performing CM activities, in the objectives, procedures, and methods for performing their CM-related duties. Examples of training include the following:

- Role, responsibility, and authority of the CM personnel
- CM standards, procedures, and methods
- CM tools and their capabilities
- Data measurement, analysis, and reporting

B.5 IMPLEMENTATION STRATEGY

The implementation strategy for establishing an effective CM program follows a phased approach in identifying, documenting, and implementing the necessary CM controls. This strategy must possess the flexibility needed to allow each HUD IT organization to define the most effective approach to (1) meet the CM objectives within their respective areas and (2) minimize the impact on the performance of HUD information systems and their missions. Listed below are activities that need to be performed when implementing the components of the CM program. The list does not imply the order in which to address the activities.

- ***Identify staff to perform CM functions.*** Identify the personnel that comprise the CM group within the division. The CM group could vary from a single part-time individual, to several part-time individuals assigned from different departments, to several full-time individuals. The size of the CM group is dependent on a variety of factors, such as number of systems, system size, and system complexity. The project can define the composition of the CM group based on the project's organizational structure. The CM group provides support for all systems within the program area.

CM support can be categorized as administrative and technical and may be divided among personnel based on skills needed. The administrative part of CM functions consists mainly of CM record keeping (i.e., documentation library maintenance, change status reports, distribution lists, CCB administrative tasks, meeting minutes, and CM-related data collection and reporting). The technical part of CM functions mainly consists of software library version control, which involves the creation and maintenance of the controlled software libraries and requires a working knowledge of the CM tools used in maintaining the libraries (e.g., ENDEVOR).

- ***Generate CM Plan.*** Each project needs a documented CMP. The CMP covers the activities to be performed, schedule of activities, assigned responsibilities for CM activities, and resources required. Depending on the project structure and size, a single CMP for the entire division may be beneficial. For a project of considerable size and complexity, a project-specific CM plan may be necessary.

- ***Establish division/project CCB.*** Establish a CCB for each IT division. The CCB should consist of representatives for all project/system stakeholders. The CCB correlates budget and technical impacts in all affected program areas before approving changes. A division CCB consists of the division's branch chiefs and is chaired by the division deputy director. The division CCB is responsible for all applications/systems managed by the division and serves as the approval authority to develop or change software, accept test results, and approve the release of software into production. Submit change requests to the CCB after an affected system-level baseline is established and/or approved. For example, if a design change is requested after the design baseline has been approved, the request has to be approved by the CCB before it is scheduled to be implemented. Appropriate provisions would be made for emergency changes.

- ***Identify and document CM standards and procedures.*** Using IT standards, create a list of CM standards and procedures needed to administer the CM objectives defined in the CMP. Identify the standards and/or procedures currently documented and in use within HUD IT. Identify the standards and/or procedures that must be developed. Assign personnel from within the project, IT organization, or both to develop the standards and/or procedures. The affected groups and the project CCB reviews and approves each procedure.

- ***Identify tools required to support CM activities.*** List the software tools currently being used to support CM activities. The software library maintenance tools currently identified are: ENDEVOR, Interactive Processing Facility (IPF), Librarian, Software Migration System (SMS), and Program Migration System (PMS). Identify other tools that are used for library control, configuration inventory and change history, and status reporting. Determine how the tools are used within the environment and determine the extent to which each tool is meeting the CM functions assigned to it.

- ***Administer required CM-related training.*** Provide CM training to selected personnel in the objectives, plans, standards and procedures, and tools required to implement the CM program. Focus the training on the following:

    - Roles and responsibilities for assigned personnel in performing CM activities

- Usage and processing of configuration control forms
- CM software library and documentation maintenance processes
- Capabilities of the software CM tools used to maintain software libraries and status reporting programs
- Types of CM reports to be produced on a regular basis

This page intentionally left blank.

Return to Title Page

Appendix C. Quality Assurance Guidelines

(To be supplied.)

This page intentionally left blank.

Return to Title Page

Appendix D. System Maintenance

This page intentionally left blank.

APPENDIX D. SYSTEM MAINTENANCE

The purpose of system maintenance is to perform the activities required to keep a system operational and responsive to users' changing needs after the system is accepted and placed into production.

System maintenance is a shared responsibility between IT and the user community. During the Operate System phase, performance and accuracy of the production system are monitored by Operations staff and the users. Any changes due to new needs or discrepancies are recorded through the appropriate change request reporting mechanism (i.e., Needs Statement [NS], Advanced Requirements Notice [ARN], or Service Ticket Action Resolution System [STARS]). Needs Statements are forwarded by the user community to the TIB for disposition; ARNs and STARs (problem tracking reports) are reported to the appropriate IT organization for disposition.

When the IT organization and the user community determine that changes are needed, approval for a maintenance release is triggered by TIB direction or by sponsor approval of an Evaluation of Request (EOR). These changes consist of corrections, insertions, deletions, extensions, and enhancements to the system hardware and software. Generally, these changes are made to keep the system functioning in an evolving, expanding user and operational environment.

System maintenance involves many of the same activities associated with system development, but it also has unique characteristics of its own. Maintenance activities are performed within the context of an existing system framework. The maintenance staff implements changes within the existing system architecture. The older the system, the more challenging and time-consuming the maintenance effort becomes, especially when minimal or no documentation exists for the system.

Maintenance activities usually are performed within shorter time frames than development efforts. A development effort may span one or more years, while a maintenance effort may be required within hours or in releases requiring from one to six months or longer to complete. Software maintenance efforts in the HUD environment, and their implementation approaches, are associated with one of the following three sizes:

- Small maintenance efforts (generally less than a staff month)
- Medium maintenance efforts (generally between one and six staff months)
- Large maintenance efforts (greater than six months)

Maintenance activities for a production system are performed by a development team of users and maintainers working together. To preserve the integrity of the production system, the team uses a separate maintenance environment so that modifications can be made and tested without affecting the production environment.

HUD uses three maintenance types to categorize and track changes to HUD applications, described below:

- **Corrective maintenance** encompasses modifications that fix application problems caused by design, logic, or coding errors. This type of maintenance is usually triggered by an explicit STAR and usually involves errors that must be investigated immediately. Examples of corrective maintenance are the following:
  - Calculations that generate incorrect totals

  - Data screens that omit a required entry or store an entry in the improper location
  - Aborted programs

- **Adaptive maintenance** generally involves modifications that are mandated in response to laws or regulations that govern HUD programs; to hardware or software changes, and/or to upgrades within HUD's technical architecture. This type of maintenance also may involve activities aimed at adding or modifying functionalities in response to new or changed requirements.

  Adaptive maintenance usually includes those activities that have been thought out and planned for future implementation. This type is usually a modification required because of software changes or a modification required because of anticipated hardware changes. Examples of adaptive maintenance are:

  - Year 2000 activities required for most of the IT software
  - Changes in software that were promulgated by new laws or regulations
  - System software upgrades such as database management systems, CASE tools, and new communications protocols

- **Perfective Maintenance** generally involves modifications that satisfy a need to improve the application in some area that is a nonfunctional requirement or to resolve a problem before the user community is affected. These activities could also be aimed at improving the maintainability of the application software. Perfective maintenance changes could be thought of as those that are "nice to have." Examples of perfective maintenance are:

  - Optimization of execution time, memory size, and file management
  - Standardization of naming conventions and development methodology

Functions and Products

Functionally, system maintenance activities mirror all or part of the development phases. However, activities are often scaled down or done in combination with each other. The activities essentially consist of successive iterations of these combined phases of the development lifecycle. All products that are produced must pass appropriate CM and QA procedures before the affected software is released into production.

Standards and Guidelines

Follow the HUD standards and guidelines when performing system maintenance activities.

Roles and Responsibilities

Throughout system maintenance, key personnel are required to perform the various tasks and activities outlined in the SDM. Table D-1 identifies the types of personnel required and the activities for which they are responsible.

Return to Title Page

Table D-1. Roles and Responsibilities for System Maintenance Personnel (1 of 2)

Role	Responsibility
Project Sponsor	Approves the maintenance schedule. Prioritizes system changes and enhancements. Prepares the budget for system changes and enhancements.
Project Manager	Prepares the maintenance schedule. Identifies the resource requirements. Develops the project WBS, schedule ,and work plan for the maintenance effort. Obtains appropriate concurrence and approvals for the schedule and work plan. Conducts a post-implementation review. Prepares a post-implementation report. Coordinates development, review, and approval of maintenance activities.
Project Development Team (User and Maintainer)	Identifies system changes and enhancements. Provides input to the project manager on required completion dates for maintenance activities. Conducts training classes, if necessary. Makes modifications and enhancements to system software, as necessary. Updates system documentation. Installs the system. Monitors system performance to ensure responsiveness to the user's needs. Develops enhancements and upgrades to software. Develops system enhancements. Conducts system and regression tests of changes and enhancements in the development environment.
Operations	Ensures that the development environment is ready. Installs the system. Performs day-to-day system operations and corrective actions, as required. Makes modifications and enhancements to system hardware, as necessary. Determines additional resource and support requirements. Monitors system performance to ensure continued responsiveness to the user community.
ADP Security	Ensures that periodic security reviews are performed on the system when in production. Reviews maintenance solutions for security impacts.

Table D-1. Roles and Responsibilities for System Maintenance Personnel (2 of 2)

Role	Responsibility
System Acceptance Test Team	Develops new test scenarios to validate changes. Performs acceptance testing activities during maintenance. Signs off acceptance of the system before production use. Conducts user acceptance testing of system changes and enhancements.
Project Database Administrator	Reviews maintenance solutions for database impact.
Project Data Administration	Reviews and approves changes to domain data models. Reviews and reconciles data models for new entities, and ensures new or changed data elements conform to HUD naming conventions.
Configuration Management	Performs change control activities. Establishes and maintains a production baseline.
Quality Assurance	Audits maintenance activities for compliance with HUD project methods, standards, and procedures.

D.1 PERFORM MAINTENANCE IMPLEMENTATION

The maintenance process for implementation of corrective, adaptive, and perfective change requests is characterized by the following activities:

- Change request initiation
- Change request identification, solution, and impact analysis
- Change request implementation
- Regression testing
- Validation and verification
- Solution acceptance
- Solution installation
- Statistics for trend analysis

Table D-2 lists the activities performed and shows how the SDM development lifecycle phases (Initiate, Define, Design, Build, Evaluate, and Operate) may be correlated to system maintenance. While phases are combined for maintenance activities, they are performed in an iterative fashion depending on the size or scale of the maintenance effort.

Table D-2. Maintenance Process Activities

Process Phase	Process Activity	Process Products (New and/or Updated)
Initiate	Change request initiation	Preliminary Maintenance Plan
Initiate	Change request identification, solution, and impact analysis	Feasibility Study, Risk Analysis, Maintenance Plan (updated)
Define, Design, and Build	Change request implementation	Security Plan; System Decision Paper; Functional Requirements Document; Data Requirements Document; System/Subsystem Specifications; Database Specifications. Program Specifications; Test Plans (unit & integration); Validation, Verification, & Test Plan
Build and Evaluate	Regression testing	
Evaluate	Validation and verification	Test and Evaluation Report, Operations Manual, User's Manual, Installation and Conversion Plan, Training Plan and Training Guide
Evaluate	Solution acceptance	Tested and accepted product
Operate	Solution installation	Product installation in production environment

Normally, system maintenance efforts are divided into the following three sizes:

- ***Small maintenance efforts:*** A small maintenance effort consists of quick reaction assignments (STARs) that can be implemented and installed with one staff month or less of effort. The assignments can address an error (corrective), a cosmetic change (perfective), or a mandated enhancement (adaptive). During a small maintenance effort, the programmer concentrates on addressing the problem and is not concerned with updating the documentation, unless the user needs updated documentation to use the software.

  Small maintenance efforts require flexibility, and the project manager must be aware of this need during planning and be responsive to it. For example, management must decide on a quick-fix maintenance effort (which may require only one day to implement) and how to combine phases for effective implementation. One approach is to have the programmer combine the Initiate, Define, Design, and Build phases into one phase. After the combined phases are complete and the software changes are tested, validated, and delivered to the user, documentation changes could be made and approved. In this case, there is no iteration of phases. If user documentation did not change, the STAR analysis becomes the documentation of the system change.

- ***Medium maintenance efforts:*** A medium-sized maintenance effort is similar to the small-sized effort, but it is generally more complex and requires one to six staff months of effort to implement and install. A medium-sized maintenance effort consists of assignments (STARs, ARNs) that address an error (corrective), a cosmetic change (perfective), or a mandated enhancement (adaptive). During a

Return to Title Page

medium-sized maintenance effort, the programmer concentrates on addressing the problem and is more concerned about updating the documentation than on a small-sized effort.

A medium-sized maintenance effort generally follows a more disciplined approach because more planning and documentation are normally required. To ensure delivery with the software modifications, produce, review, and approve user documentation of the modifications before the maintenance product is put into operation.

- ***Large maintenance efforts:*** A large-sized maintenance effort requires an implementation and installation effort of more than six staff months to address one or more errors (corrective), cosmetic changes (perfective), or mandated enhancements (adaptive).

  A large-sized maintenance effort follows a disciplined approach because planning and documentation are required. It is recommended that the project manager and the development team follow the SDM development lifecycle phases, without combining them, for this type of maintenance.

D.1.1 Change Request Initiation

Initiation of the maintenance process is the receipt of one or more software change requests (Needs Statement, ARN, or STAR) for a system. In general, the requests should contain the problem or need for change, type of maintenance requested (corrective, perfective, or adaptive), and priority level. Assign a project development team to analyze requests. If a change request is a problem identified by a STAR, determine if the problem is due to a misunderstanding of procedures or an actual processing malfunction. Also, determine if the problem is already being resolved in response to another STAR. Provide a needs assessment, and generate a preliminary feasibility study for change requests.

Concurrently, the project manager defines a preliminary schedule, resource profile, WBS, and budget and documents the preliminary information in a maintenance plan.

D.1.2 Change Request Identification, Solution, and Impact Analysis

This step localizes the change request to the precise parts of the affected software. Develop various change alternatives to satisfy the change request. Perform an impact analysis to evaluate the consequences to the system of each alternative, and select a solution. The project manager updates the maintenance plan to reflect the process activities and schedule, based on the scale of the effort to implement the solution.

D.1.3 Change Request Implementation

The development team performs the unit modifications (design and code), generates unit and integration test plans, and performs unit and integration testing. Based on the scale of the effort, the development team updates the documentation specified in the maintenance plan. The acceptance test team updates the VV&T Plan with the new test procedures and expected results. QA monitors the implementation process for adherence to processes and standards.

D.1.4 Regression Testing

The development team tests the new or modified components for compatibility with the existing system. Use the VV&T Plan that was generated during the development of the system. Choose the tests from the VV&T Plan that validate the functionality being modified by, or being affected by, the maintenance change. Use a test environment that duplicates the operational environment. CM configures the test environment.

D.1.5 Validation and Verification

This step is an independent verification and validation process that ensures the quality of the modified components. The acceptance test team, independent of the project development team, executes the new test procedures that validate the modifications and verifies that all of the change request requirements have been met. Additionally, the acceptance test team performs independent regression testing to verify that the maintenance change has not affected existing functionality. Use the original VV&T Plan to choose a limited set of regression tests. Document and forward any discrepancies to the maintainers for resolution. QA monitors test results and test plans. Make modifications to the User's Manual, Operations Manual, and Installation and Conversion Plan, as necessary. Check all modified documentation for conformity to requested changes.

D.1.6 Solution Acceptance

The requester of the change performs solution acceptance. The requester checks the test results to verify that the solution fulfills and resolves the requested change. QA verifies that only authorized work was completed.

D.1.7 Solution Installation

This is the step during which the modified software is placed into a staging area or is installed in the production environment. A production control unit normally replaces the old version with the new one.

D.1.8 Statistics for Trend Analysis

Collected and analyze statistics for maintenance ARNs and STARs. Focus the analysis on why a change was made, the type of change made (corrective, perfective, adaptive), and which areas were affected by the change. Reasons why a change was made could be a change in HUD law, hardware upgrade, support software upgrade, or new requirement on the system. Areas affected by a change could include the program, hardware, environment, database, data reports, documentation, and users. The data can be used to determine the impact on resources required to enhance the HUD system.

D.2 PERFORM POST-IMPLEMENTATION REVIEW

After the system is installed and operational and initial problems or "bugs" have been resolved, conduct a post-implementation review. The level of effort and depth of this review can vary. For example, the data collection and analysis activity can be modified in relation to the size and scope of the maintenance effort.

Return to Title Page

D.2.1 Determine Time for Evaluation

Do not hold this review immediately after system implementation; postpone it until adequate and accurate data can be gathered regarding system qualities and characteristics. Schedule the review after all users are thoroughly familiar with the automated system and all of the software problems have been resolved. This review should take place approximately 90 days after implementation and again in six months.

D.2.2 Conduct Evaluation

Conduct the post-implementation evaluation using interviews of operations personnel and user representatives. Review project development and operation documentation. Determine the following:

- Does the system meet the stated needs of the user? Are those needs still applicable? How much longer will the system meet those needs?
- Were the final project staffing and cost estimates within the original estimates?
- Have requirements changed, making the hardware, system software, and application software obsolete?
- Have revisions to the system been made only as necessary, and are they being accomplished effectively and efficiently?
- Was the project well controlled and documented?
- Is the system responsive to users? Do the system functions meet user expectations? Are the system interfaces, screens, and documents user friendly?
- Is the data in the system accurate and nonredundant?
- Are the controls (e.g., access, data validation, run-to-run) in the operational system adequate?
- Can benefits be identified in specific categories (e.g., cost avoidance, cost reduction, opportunity yield, direct revenue yield, intangibles)?
- Are cost savings being met?
- Have regulations or laws changed? Will these changes impact the volume of work?
- Because of automation, have staff duties and procedures changed? Will these changes impact the benefit/cost ratio?
- Does the system receive adequate support from operations personnel?

D.2.3 Formulate Recommendations

The objective of the review is to determine if the system has achieved or surpassed the system's goals. If it is determined during the review that the desired return on investment has not been achieved, develop recommendations for achieving the desired return.

D.2.4 Document Evaluation and Recommendations

Document the review findings to reflect the pre- and post-automation results. Develop documentation as a standardized means of capturing data and providing a comparison of anticipated and actual benefits derived from the automation effort.

D.2.5 Present Evaluation for Management Review

Forward copies of the review findings and recommendations to the management officials responsible for acquisition, development, and implementation of the system. While analyzing these results and recommendations, management should ask the following questions:

- Has work flow improved since the introduction of automation? For example, have unproductive or repetitive tasks been eliminated?
- Has the paperwork improved in such areas as appearance, format, usefulness, accuracy, and completeness?
- If processing time has been and continues to be tracked, has this time decreased?
- Is the new system achieving the estimated goals planned before the new system was implemented?
- Has feedback from users and clients been evaluated and incorporated to improve the effectiveness of the system?

D.3 PERFORM CHANGE CONTROL ACTIVITIES

Perform change control activities whenever any products being baselined under the project's CM function are revised. Change control activities include verifying changes made to the products, assigning a new version number to the revision, updating the logbook with the change information, updating the central library with the new version, distributing copies of the new version, and archiving the old version.

D.3.1 Verify Changes Made to Product

Review the updated product to ensure that changes have been made as described in the supporting documentation. The supporting documentation may be comments received from document reviews. The supporting documentation for software may be software change requests or discrepancy reports generated during testing.

D.3.2 Assign New Version Number

Assign a new version number to each updated technical deliverable (hardware or software configuration item) and document deliverable. The version number must follow conventions established by the project's CM function, within HUD guidelines, and enable monitoring of the distribution of official copies and of the removal of outdated versions from the workplace.

D.3.3 Store Approved Product in Central Library

After updating a baselined product, store the updated version in the project's central library. The library may be a physical storage space for hard copy documentation and program listings or an on-line subdirectory or partitioned data set where programs are stored. The central repository must be maintained in a manner that facilitates auditing the new version and all prior versions of a baselined product. To protect the integrity of the project's products, the ability to move CIs into and out of the library should be limited to as few staff members as possible and strict procedures should be established in the CMP and adhered to. These procedures will preventing old versions of CIs from overwriting newer versions and the same product from being updated concurrently.

D.3.4 Update Inventory Log

Ensure that updated product information is recorded in the inventory log. This information should include the following details:

- Title of the product
- Latest release date
- New version number
- Name and version or model numbers of the software and hardware used in development of the new version of the product
- Organization responsible for updating the product (usually the sponsoring organization)
- The product's distribution list
- Updates to the central library

D.3.5 Distribute Copies of Products as Required

After an approved, updated version of the product is stored in the project's central library, distribute copies of the updated product or change packages following the same distribution channels and procedures used for the original product. Institute a plan to ensure the prompt removal of outdated versions from the work area.

D.3.6 Archive Old Version of Products

Outdated versions of all products should be archived and retained for the period of time required to comply with HUD standards and guidelines. Old versions of software may be moved to tape or may be moved to a separate subdirectory or data set; store only the latest version in the main software library to prevent old versions from being tested, reworked, and released for production.

Appendix E. Document Review Checklist

This page intentionally left blank.

APPENDIX E. DOCUMENT REVIEW CHECKLISTS

Appendix E contains review checklists for the project products that may be tailored into a delivery. The checklists are used during generation of the document products and during review of the products. The products are listed below:

- Needs Statement
- Feasibility Study
- Benefit/Cost Analysis
- Risk Analysis
- System Decision Paper
- Project Plan
- Functional Requirements Document
- Data Requirements Document
- System Security Plan
- System/Subsystem Specification Document
- Database Specifications
- Program Specification Document
- Test Plan
- Validation, Verification, and Test Plan
- Training Plan
- User's Manual
- Operations Manual
- Maintenance Manual
- Installation and Conversion Plan
- Training Guide
- Test Results and Evaluation Report

Return to Title Page

Needs Statement Review Checklist

#	Objective	Review Questions	Tools and Techniques
1.	Needs Statement/ Advanced Requirements Notice (NS/ARN) clearly defines the need and justifies implementing that need.	Have the sponsor and development staff reached a common understanding of the information problem described in the NS/ARN? Does the NS/ARN clearly describe the user's information problem in nontechnical terms? Are the user's needs stated in measurable terms? Is the NS/ARN adequate and in compliance with all applicable HUD standards?	Interview sponsor and development staff to ensure each has the same understanding of the need. Conduct review.
2.	User’s management is participating in the Initiate Project phase.	Is the user’s management fully involved in the activities of the phase?	Interview user management.

Feasibility Study Review Checklist

#	Objective	Review Questions	Tools and Techniques
1.	Feasibility Study is clearly defined and documented.	Was the feasibility of the proposed system determined by a reasonable method from among the various alternatives? Are there no adequate alternatives available to meet the user's needs without building a new system? Are all the assumptions made in arriving at the proposed solution valid? Can the data required by the system be collected with the desired degree of reliability? Can the functions required by the user be accomplished by the proposed system? Is the technology available to develop and operate the proposed system? Does the proposed solution solve the business problem? Is the Feasibility Study documentation adequate and in compliance with all applicable HUD standards?	Ensure that the alternatives include the information and analysis prescribed by applicable HUD standards and guidelines. Confirm with the users that there are no existing systems that will meet their needs. Review the document to ensure that all considerations have been adequately addressed.

Return to Title Page

Benefit/Cost Analysis Review Checklist

#	Objective	Review Questions	Tools and Techniques
1.	Benefit/Cost Analysis includes all of the cost and benefit considerations associated with the initiation, operation, and maintenance of the proposed project	Has the cost of developing and operating the system been calculated? Has the cost of developing and operating manual segments been calculated? Could characteristics of the proposed system cause the actual cost to vary significantly from the projections? Have the benefits of the system been calculated? Has the useful life of the system been calculated? Has the cost of compliance been measured against the proposed benefits? Are there potential system characteristics that could cause the benefits to vary significantly from the projected benefits? Is the Benefit/Cost Analysis documentation adequate and in compliance with all HUD standards?	Review the document to ensure that the analysis has been conducted in accordance with HUD's standards and procedures.

Risk Analysis Review Checklist

#	Objective	Review Questions	Tools and Techniques
1.	Internal control and security vulnerabilities, as well as the magnitude of associated threats, have been determined.	Have the sponsor and developer reached an understanding of the proposed project's risks? Does the analysis assess the project's probability of success? Have potential problem areas been identified? Has an assessment been made on technological expertise required and its availability within the organization? Do the security controls for the protection of data meet the requirements of the Privacy Act of 1974? Have the appropriate procedures for the disposition of sensitive material been identified? Have alternate operating procedures been identified should the system fail? Is the Risk Analysis documentation adequate and in compliance with all applicable HUD standards?	To ensure that the results are reasonable, assess the makeup and performance of the risk management team, the methodology used, and the completeness of the results against the vulnerabilities and risk analysis methodology described in HUD ADP security standards and guidelines.

System Decision Paper Review Checklist (1 of 2)

#	Objective	Review Questions	Tools and Techniques
1.	The System Decision Paper includes all information needed by user management to make a decision on action to be taken for the proposed project.	Is there a consensus among the user and developer concerning the recommended, alternative benefits/costs and the technological aspect of the proposed project?	Interview users and developers to ensure agreement regarding the proposed project.
		Is the System Decision Paper adequate and in compliance with all applicable HUD standards?	Compare the structure and content of the System Decision Paper to the structure, format, and content described in HUD standards and guidelines.
2.	The System Decision Paper includes all information needed by user management to make a decision on action to be taken for the next phase of the automated information system.	Is there a consensus among user areas (Program Offices) and designers concerning the recommended alternative, the benefits/costs, and the technological aspects of the systems implementation approach?	Use structured interview techniques.
		Does the System Decision Paper include all of the essential information on the automated information system including the following? a) Mission need b) Risks c) Alternatives d) Benefits/costs e) Management Plan f) Supporting rationale for decisions g) Affordability in terms of projected budget and out-year funding h) Consistency with long-range plans and other Information Resource Management (IRM) initiatives	Compare the document's structure, format, and content to the applicable HUD standards and guidelines.

System Decision Paper Review Checklist (2 of 2)

#	Objective	Review Questions	Tools and Techniques
3.	The system decision continues to be supported by documents.	Has the System Decision Paper been updated? Is there consistency between the System Decision Paper and all other Define System phase document deliverables? Is there consistency between the System Decision Paper and all other Design System phase document deliverables? Is there consistency between the System Decision Paper and all other Build System phase document deliverables? Is there consistency between the System Decision Paper and all other Evaluate System phase document deliverables?	Obtain and review a copy of the updated System Decision Paper.

Project Plan Review Checklist

#	Objective	Review Questions	Tools and Techniques
1.	A Project Plan has been developed that specifies the strategy for managing the software development effort.	Does the Project Plan identify the strategy for managing the development effort?	Compare the attributes of the Project Plan to other projects of equal size and complexity to validate the reasonableness of estimates and milestones.
		Does the Project Plan identify goals and activities for all of the phases and subphases, and does the plan include milestone dates and resource estimates?	Compare the Project Plan and subplans to HUD standards and guidelines.
		Are the subplans, included within the Project Plan, accurate and complete?	
2.	The revised Project Plan is current and provides the direction needed to manage the project effectively and efficiently.	Does the Project Manager confirm that the plan is up to date and being followed?	Compare the status of completed documents to the document's status as indicated by the Project Plan.
		Does the plan provide adequate information to adjust project direction as appropriate to ensure the project will be completed on time, within budget, and will produce the expected deliverables?	Verify that the plan is accurate; then, through interviews with the developers, ensure that problems in their work are appropriately addressed by project management.
3.	There is consistency within the plan itself.	Have the various plans included in the Project Plan (i.e., QA, System Support, Configuration Management, Security, Internal Audit, etc.) been reviewed to ensure that they are in agreement with each other?	

Functional Requirements Document Review Checklist

#	Objective	Review Questions	Tools and Techniques
1.	A definition of existing and new information requirements is specified in detail.	Are the existing and new information requirements complete? Are they specified in enough detail to permit test data generation in subsequent phases for compliance verification?	Verify the information attributes in the Functional Requirements Document to the definitions in the data dictionary.
2.	The level of service necessary to achieve the processing objective is defined and documented.	Are the information requirements complete and consistent with standard data processing definitions? Has a desired percentage of uptime been specified? Has the response time for each transaction been specified? Is the response time reasonable, based upon user department needs?	Interview key user personnel to validate that the specified service levels are adequate.
3.	Detailed requirements for processing steps are defined and documented.	Are review processing requirements adequate and prepared in accordance with management policies?	Interview user personnel to validate information requirements.

Data Requirements Document Review Checklist

#	Objective	Review Questions	Tools and Techniques
1.	All input and output requirements should be defined and documented.	Does the documentation for input requirements of the new system include the following? a) Editing and validation requirements b) Input or update authorization c) Establishment of appropriate control totals d) Required precision for each quantitative field e) Time requirements for the entry of each transaction f) Requirements for handling inaccurate error identification/correction or incomplete data g) Rules for authorizing each of the key transactions h) Specified retention requirements for input and output data i) Description of input and output forms, transactions, sources, and volumes	Interview user personnel to validate data requirements. Validate that information requirements are complete. Validate attributes of data to those specified by appropriate regulations. Review the document to ensure that HUD standards have been met.

Return to Title Page

System Security Plan Review Checklist

#	Objective	Review Questions	Tools and Techniques
1.	All system security requirements are defined and documented.	Does the System Security Plan include security, control, and privacy issue requirements? If so, are those requirements adequate? Do the requirements address previously defined risks?	Identify the control techniques needed to minimize vulnerabilities for the proposed system.
2.	The System Security Plan identifies sensitive and critical data and assets and determines how those items should be controlled during computer processing.	Does the System Security Plan indicate the data and asset sensitivity/criticality protection requirements?	Review the appropriate legislation (e.g., Privacy Act of 1974, Freedom of Information Act) to validate that the types of transactions, data, and assets governed by the system will be adequately protected.

Return to Title Page

System/Subsystem Specification Document Review Checklist

#	Objective	Review Questions	Tools and Techniques
1.	Sufficient data processing and security controls are incorporated in the detailed design to ensure the integrity of the system.	Have the detailed design specifications been reviewed? Have the system controls to be built into the system to evaluate the adequacy of those controls been identified?	Review the documented controls in the detailed design. An absence of controls at risk points indicates potential control weakness.
2.	Rules for authorizing transactions are defined and documented.	Have the methods for authorizing each transaction been documented? Is each method reasonable?	Use a structured interview technique to determine that the rules for authorizing paper transactions have been defined and that all electronically originating transactions have been identified.
3.	Documentation suitable for use as audit trails are incorporated into the detailed design.	Do the audit trail specifications include both the automated and manual segments? Is the audit trail adequate to trace transactions from the source documents to control totals and back to supporting transactions?	Prepare a document flow diagram to pictorially show the flow of documents, including electronic documents, and to show who is responsible for these documents and where the documents are stored. The specific audit trail would be illustrated through notation on the document flow diagram and by references to the specific documents, both manual and electronic.

Return to Title Page

Database Specifications Review Checklist

#	Objective	Review Questions	Tools and Techniques
1.	The database specifications provide the correct architectural solution to meet the documented requirements.	Does the documented database solution provide the information needed to meet the objectives? Does the document describe the functions and performance requirements? Are the performance requirements described in terms of accuracy, validation, timing, flexibility, and operating environment? Do the Database Specifications address storage and design considerations?	Establish a team of technical peers to review the design to ensure that it meets the system's requirements.

Program Specification Document Review Checklist

#	Objective	Review Questions	Tools and Techniques
1.	The program specifications provide the correct architectural solution to meet the documented requirements.	Does the document specify the program operating environment? Does the document specify the program requirements? Does the document specify the program design characteristics? Does the document describe the function and performance requirements?	Establish a team of technical peers to review the design to ensure that it meets the system's requirements.
2.	Sufficient data processing and security controls are built into the detailed design to ensure the integrity of the program.	How are the controls specified in requirements designed into the program?	Review the documented controls in the detailed design. An absence of controls at risk points indicates potential control weakness.

Test Plan Review Checklist

#	Objective	Review Questions	Tools and Techniques
1.	A Test Plan is developed and documented for unit tests.	Does the Test Plan include detailed specifications, descriptions, and procedures for testing all units? Does the Test Plan exercise all decision points of the software units? Are all test drivers and stubs identified? Does the Test Plan include test data reduction and evaluation criteria? Do the plan's test procedures identify expected results and exercise erroneous data conditions?	Prepare a function/test matrix that lists all application functions (i.e., requirements) on one axis of the matrix and then cross-references them to all tests included in the Test Plan.
2.	A test plan is developed for testing integrated components.	Is every function in the Functional Requirements Document covered in the Test Plan? Has a test hierarchy for integrating and testing components/units been identified? Is each possible test given the resource and environmental constraints? Can each test be completed in the scheduled time, allowing for a reasonable number of delays, etc.? Have all of the necessary personnel been identified for operating test equipment and the system itself? Have all of the necessary equipment and software (external programs) been identified? Have data been identified and structured to handle erroneous conditions and to test all program interfaces? Have all the necessary "soft" materials been identified (e.g., preset files and databases)?	Prepare a functions/test matrix that lists all application functions on one axis and then cross-references them to all tests included in the test plan. This provides proof that the Test Plan is complete.

Validation, Verification, and Test Plan Review Checklist

#	Objective	Review Questions	Tools and Techniques
1.	A VV&T Plan is developed and documented.	Does the plan adequately validate the system requirements? Does the plan include detailed specifications, descriptions, and procedures for testing all systems? Does the plan include test data reduction and evaluation criteria? Does the document define testing techniques and tool selection information? Does the document describe measurements and constraints?	Prepare a functions/test matrix that lists all application functions on one axis and then cross-references them to all tests included in the VV&T plan.
2.	The VV&T Plan is finalized and documented.	Is every function in the Functional Requirements Document covered in the VV&T Plan? Is each possible test given the resource and environmental constraints? Can each test be completed in the scheduled time to allow for a reasonable number of delays, etc.? Have all of the necessary personnel been identified for operating test equipment and the system itself? Have all necessary equipment and software been identified? Have all the necessary “soft” materials been identified (e.g., preset files and databases)?	Update the functions/test matrix that lists all application functions on one axis and then cross-references them to all tests included in the test plan. This provides proof that the VV&T Plan is complete.

Training Plan Review Checklist

#	Objective	Review Questions	Tools and Techniques
1.	Training requirements for all job categories of personnel that will interface with the system have been determined.	Have the training needs of the following classes of personnel been identified? a) Users b) Developers c) Testers d) Operators e) Managers Have the prerequisites for individuals to receive training been identified, and has a strategy for prerequisite training been developed?	Establish a team of technical peers to review the training plan to ensure that it adequately provides for the needs of the system's personnel.
2.	A complete schedule for training has been developed.	Does the training schedule identify the following? a) Training material development b) Training course development c) All training classes	
3.	The training will be effective.	Have the training methods been identified? Has CBT been incorporated? Will training include hands-on use of the system?	
4.	A plan has been developed to train the users and system personnel in the use of the system.	What are the different types of system users (operations, testing, end user, systems support staff, etc.) that may require separate, special training? Are training courses to be held only once, or will regular courses be required to train new users? Would a computer-aided instruction program be appropriate for any of the training? Is a system simulator necessary for training? If so, how will the simulator be constructed from the real system? Will suitable trainers be available?	

Return to Title Page

User's Manual Review Checklist

#	Objective	Review Questions	Tools and Techniques
1.	The manual for users is prepared and adequately documented.	Is the User's Manual written in non-ADP terminology? Does the User Manual adequately address the following? a) System functions b) System use c) Preparing input data and parameters d) Interpreting output results e) Full description of the application f) User operating procedures g) User responsibilities related to security, privacy, and internal controls h) Error detection and correction i) Error recovery j) File query procedures	Review document, and ensure that it contains information to fulfill each review question.

Operations Manual Review Checklist

#	Objective	Review Questions	Tools and Techniques
1.	The manual for operators is prepared and adequately documented.	Does the Operations Manual include, for each job step, the following information? a) Program function b) Hardware requirements c) Explanation for all console messages together with appropriate operator response d) Output creation and its disposition e) Proper identification of output file labels f) Appropriate restart or notification procedures specified for error or failure conditions	Review document, and ensure that it provides information to adequately answers all review questions.

Return to Title Page

Maintenance Manual Review Checklist

#	Objective	Review Questions	Tools and Techniques
1.	A Maintenance Manual is prepared with adequate information on projected maintenance needs and problems.	Does the Maintenance Manual project the following? a) Periodic software needs b) Periodic hardware needs c) Possible problem areas Does the manual provide the maintenance programmer with the following information? a) Information and source code necessary to understand the programs b) An overview of the architecture/structure of the system c) Maintenance guideline procedures d) The design of internal control and security procedures so that they can be individually maintained	Review manual, and ensure that adequate information is contained in the manual for maintenance to be conducted.

Installation and Conversion Plan Review Checklist (1 of 2)

#	Objective	Review Questions	Tools and Techniques
1.	An Installation and Conversion Plan is prepared and adequately documented.	Does the Installation and Conversion Plan explain how to do the following? a) Install newly developed software b) Activate security procedures c) Interconnect the software with other related software packages d) Convert existing software Are the parts of the plan that are directed towards staff personnel presented in nontechnical language? Are the parts of the plan directed toward operations personnel presented in suitable terminology? Do the Installation procedures in the plan explain the following? a) Necessary building modifications b) Necessary security procedures c) Interfaces between the system's software and other related software packages and systems d) Procedures for installing the software, hardware, and peripheral equipment in the operating environment	Use a structured interview technique to determine if all procedures for installation and conversion of the system have been identified.
2.	The installation and conversion is smooth.	Have all procedures for implementing the new system been identified, and is adequate detail provided? Does the plan include a schedule for performing the installation? Does the plan include a schedule for performing the data conversion?	

Installation and Conversion Plan Review Checklist (2 of 2)

#	Objective	Review Questions	Tools and Techniques
3.	The plan provides adequate information to organizations affected by implementation of the system.	Are the parts of the plan that are directed toward staff personnel presented in nontechnical language? Are the parts of the plan that are directed toward operations personnel presented in suitable terminology? Are all organizations involved in the installation and conversion identified in the plan, and are their roles clearly defined? Do all parties involved with the installation and conversion concur with the procedures?	

Training Guide Review Checklist

#	Objective	Review Questions	Tools and Techniques
1.	The Training Guide is complete and covers all types of training classes to be provided.	Will separate Training Guides be needed for the following? a) Management (overview) b) System Manager c) User system support staff d) Application users If so, have they been prepared? Will the User's Manual be available as a supplement to a Training Guide either before or during the training course?	

Return to Title Page

Test Results and Evaluation Report Review Checklist

#	Objective	Review Questions	Tools and Techniques
1.	Testing is conducted according to the Validation, Verification, and Test Plan.	Are the tests conducted the same as the tests described in the updated VV&T Plan? Have any deviations from the VV&T Plan been documented in the report?	Compare test report to the VV&T plan.
2.	All functions have been tested.	Does the report clearly describe each function tested? Does the report indicate those functions that passed testing? Does the report indicate those functions that failed to pass testing?	Review Test Results and Evaluation Report.
3.	All procedural, operational, and performance testing is complete.	Does the report clearly describe each test executed to validate the following? a) Procedural b) Operational c) Performance objectives Does the report indicate those test objectives that the system passed? Does the report indicate those test objectives that the system failed to pass?	Review Test Results and Evaluation Report.
4.	Security considerations are taken into account.	Does the report indicate those tests executed to demonstrate the system’s security capabilities? Does the report indicate any security considerations taken into account while the system acceptance tests were executed? Does the report indicate anyde a general statement of the system's readiness to be placed into production? Does the report summarize all system deficiencies found during system acceptance testing? Does the report itemize suggested improvements for the system?	Review Test Results and Evaluation Report.

Return to Title Page

Appendix F. Work Breakdown Structure

This page intentionally left blank.

APPENDIX F. WORK BREAKDOWN STRUCTURE

Purpose

This appendix provides an introduction to the fundamentals of the HUD IT work breakdown structure (WBS) and its application to the work efforts defined in the HUD IT environment. This guidance is consistent with HUDCAPS Project Cost Accounting Subsystem (PCAS). The information will be expanded in a subsequent edition of the SDM. In the meantime, the information in this appendix will be supplemented with training for managers, as needed.

Overview

User needs in the HUD environment are typically organized into projects for implementation. Projects consist of one or more components such as studies, system development efforts, system maintenance efforts, or other types of computer-based work. For the purposes of planning, scheduling, and controlling a project, it is good practice to organize the work of project components into a structure of well-defined work efforts and work elements. The structure is defined as the work breakdown structure (WBS) for the project.

To create a project WBS, the project manager begins with the new standard WBS template showing the generic activities for defining a project or project component. The template is generated either manually or automatically, depending on available tools. The advantage of using a standard WBS is that it helps when performing the following activities:

- Ensure completeness of the work plan for the project or component. The standard WBS template serves as a planning checklist to identify all potential project-related work. The template enables the project manager to readily draft a project WBS, thereby saving time and reducing the likelihood that necessary project work will be inadvertently omitted. The project manager then tailors the standard WBS at the lowest level by deleting elements as required. The result is a WBS suited to the particular project or project component.

- Provide a chart of sub-projects. The standard WBS can help provide organized methods for allocating the project budget, assigning budget responsibility, and collecting actual project costs. Since the WBS identifies all project-related work, the budget can be distributed across the WBS down to the level in the WBS hierarchy appropriate for that particular project. Sub-projects can then be established at the proper levels in the WBS. A sub-project is a control point at which actual costs are accumulated and compared to budgeted costs for the work performed. A sub-project is also a point of responsibility. Usually, an individual on the project or a project component is assigned responsibility for all work (and budgets) within a sub-project.

- Facilitate comparison between projects or components such as systems. Since the standard WBS provides a uniform structure for collecting actual expenditures on system development projects, it facilitates comparison of performance across similar projects. It also allows the project manager to compare estimates for new projects and systems with actual costs and efforts from prior projects and systems. Such comparisons may uncover potential problems with project estimates.

F.1 ASSUMPTIONS FOR A STANDARD HUD IT WBS

Make the following assumptions for applying a standard HUD IT WBS to projects:

- A project is characterized by the work for one or more components, such as a standalone study, some form of computer-based implementation, or the development or maintenance for one or more systems to meet a user's need.

- A project may consist of multiple work efforts for multiple systems.

- A system consists of the following types of work efforts:

  - Development
  - Corrective maintenance
  - Adaptive maintenance
  - Perfective maintenance

- The fundamental units for capturing and rolling up cost data consist of the following elements:

  - Job # (Also referred to as PCAS # in some systems). (Identifies the user's need to be satisfied).
  - System ID (identifies the system(s) to be developed or modified)
  - Project (divides the total job into one or more logical or workable levels of activity)
  - Sub Project or task (spending occurs against subprojects, which then "rolls up" to the project level)

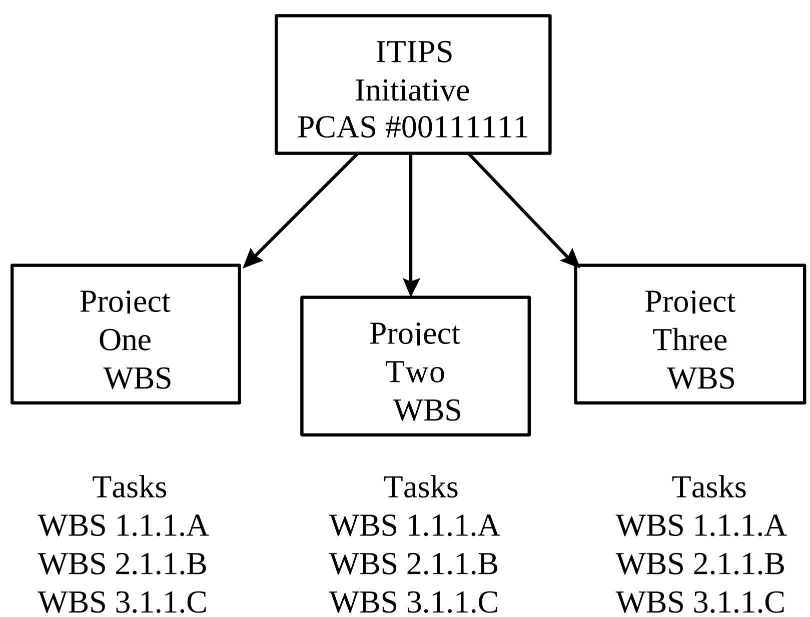

Fig. F-1 Fundamental WBS Elements

- The standard HUD IT WBS has one set of 10 cost accounts, which equate to the 10 IT Life Cycle Cost Elements. Each cost account lists the generic tasks that apply to that type of cost element.

- A. Project Initiation/Planning
- B. Requirements Definition
- C. System Design
- D. Software Acquisition
- E. Hardware/Infrastructure Acquistition
- F. Development/Perfective Maintenance
- G. System Integration and Testing
- H. Installation and Deployment
- I. System Operations
- J. Corrective and adaptive Maintenance

- Each instance of the WBS can be rolled up to get total project (and Initiative) status.

F.2 APPLYING HUD IT WBS TO DEVELOPMENT AND MAINTENANCE EFFORTS

The generic structure of the HUD IT WBS, illustrated in Figure F-2, is applicable to all development and maintenance efforts. The dictionary for the HUD IT WBS elements is provided in Section F.3.

F.2.1 WBS for Development Effort

For a development effort, use any of the first 8 (A - H) cost elements of the generic WBS (see Figure F-2) as the template from which to build the project WBS. Depending on the scope of the effort, the project manager may choose a subset of the cost elements defined by the generic WBS.

F.2.2 WBS Tailoring for Maintenance Efforts

Determine the size of the maintenance effort before considering a WBS for software maintenance efforts. Normally, software maintenance efforts can be divided into three categories based on size:

- Large (efforts requiring more than six staff months)

- Medium (efforts requiring from one to six staff months)

- Small (efforts requiring less than one staff month)

Establishing WBS for Large Maintenance Efforts

A large-sized maintenance effort (corrective, perfective, or adaptive) consists of a planned maintenance effort that addresses either an error (corrective), a cosmetic change (perfective), or a mandated enhancement (adaptive) that requires an implementation and installation effort of more than six staff months. The effort is defined by a Needs Statement. A large-sized maintenance effort follows a disciplined approach requiring planning and documentation. For this type of maintenance, a project manager could use a full set of cost accounts, similar to a development effort for Perfective Maintenance, or as few as a single cost element for Adaptive or Corrective Maintenance . Cost elements I and J would be are used for Adaptive and Corrective Maintenance initiatives.

Return to Title Page

Establishing WBS for Medium Maintenance Efforts

A medium-sized maintenance effort consists of a quick-reaction assignment that may be corrective, perfective, or adaptive and that can be implemented within one to six staff months of effort. The effort may be defined by one or more Cost Elements. During a medium-sized maintenance effort, the programmer addresses the problem and updates the documentation to reflect modifications. Because of the planning and documentation normally required, a medium-sized maintenance effort generally follows a disciplined approach. A recommended, but not mandatory, approach for establishing a WBS for this type of effort is as follows:

- Use a minimal number of cost accounts. (Use only those that enable a project manager to control the maintenance effort.)

- Use cost elements F, I or J, depending on the type of maintenance effort. (Consider collecting all costs for project management, quality assurance, configuration management, and other direct costs [ODCs] in one cost account.)

A small-sized maintenance effort consists of a quick-reaction assignment that may be corrective, perfective, or adaptive and that can be implemented and installed within one staff month or less of effort. The effort may be defined by the same Cost Elements identified above. A small maintenance effort is similar to the medium effort but is generally less complex and requires less effort to implement and install. During a small maintenance effort, the programmer usually focuses only on the problem and is not concerned with updating the documentation, unless required by the software user. Most documentation (if any) can be updated after the software modification is installed and operational. A recommended, but not mandatory, approach for establishing a WBS for a small effort is as follows:

- Use the minimal number of cost accounts. (Use only those that enable a project manager to control the maintenance effort.)

- Use cost elements F, I or J, depending on the type of maintenance effort. (Consider collecting all costs for project management, quality assurance, configuration management, and other direct costs [ODCs] in one cost account.)

"WBS Life Cycle
Cost Estimates For I

Fig. F-2 WBS Template and Worksheet

F.3 WBS DICTIONARY

The following describes the content of each WBS element.

WBS Number	Activity	Description
A. Project Initiation/Planning		
1.1.1	Perform Project Tracking and Control	Provides adequate visibility into actual progress so that management can take effective actions when the project's performance deviates significantly from the software plans. Project tracking and control involves tracking and reviewing the project's accomplishments and results against documented estimates, commitments, and plans, and adjusting these plans based on the actual accomplishments and results. Progress is determined by comparing the actual project size, effort, cost, and schedule to the plan when selected work products are completed at selected milestones.
1.1.2	Perform Needs Assessment	Gather information about performance discrepancies and using information to make decisions about priorities. The priorities that must be determined are the goals for the problem-solving effort. The data collected should provide a basis for stating the goals. The problem is converted from needs statements to what should be achieved by the solution.
1.1.3	Develop Project Plan	Document the activities to be performed during the software development life cycle. This plan will also explain how, by whom, and when these activities will be performed and verified. The plan should include estimating, supervision and implementation as well as the SDM activities. A project plan should also include risk identification, assessment and mitigation strategies.
1.1.4	Create Project Schedule	This document is developed to track project task, when specific tasks are due and who is responsible for completing the specified assignments.
1.1.5	Perform Feasibility Study	The purpose of a feasibility study is to investigate the options that are available to the architecture of the system from the mid to longer-term future.
1.1.6	Perform Cost/Benefit Analysis	A cost-benefit analysis is used to determine the most cost-effective approach or combination of approaches for development and operations of a system.
1.1.7	Perform Risk Analysis	Risk analysis involves the selection of cost-effective safeguards that will reduce security risks for a system to an acceptable level. The analysis includes identifying system and baseline security requirements and vulnerabilities; estimating potential losses that may result from vulnerabilities and the damage from the occurrence of threats; identifying countermeasures; and documenting the results of project approval.

WBS Number	Activity	Description
1.1.8	Develop Configuration Management Plan	A Configuration Management Plan identified policies and procedures for the configuration management activities necessary to support a project in the operation and maintenance of its systems. The plan establishes the configuration management policies associated with the software development, maintenance, and configuration management of the entire system. The plan also defines the procedures for requirements definition/implementation, change request disposition and version control.
1.1.9	Develop Quality Assurance Plan	The Quality Assurance Plan is developed to documents the process being used by the software project and the products being developed. The plan outlines the reviews and audits process conducted on the products to verify that they are complying with the applicable procedures and standards.
1.1.10	Develop Risk Management Plan	A Risk Management Plan identifies areas of possible risk and outlines corrective actions to be taken to reduce the risk to an acceptable level. It includes those activities required to assure efficient management of the acquisition process and individual program. Risk Management covers disciplines such as systems test evaluation, and quality assurance.
B. Requirements Definition		
2.1.1	Perform Project Tracking and Control	Provides adequate visibility into actual progress so that management can take effective actions when the project's performance deviates significantly from the software plans. Project tracking and control involves tracking and reviewing the project's accomplishments and results against documented estimates, commitments, and plans, and adjusting these plans based on the actual accomplishment and results. Progress is determined by comparing the actual project size, effort, cost, and schedule to the plan when selected work products are completed at selected milestones.
2.1.2	Define User Requirements	User Requirements are obtained from interactions with the users. The requirements are gathered to identify their needs and defines the system functionality that is required.
2.1.3	Define System Requirements	System Requirements are obtained from the customer requirements as well as system objectives (Needs Statement); these are used as the basis for defining the required functionality and capabilities of the system.

WBS Number	Activity	Description
2.1.4	Define Functional Requirements	Functional Requirements entail developing a detailed set of operational, interface, performance, functional, and control requirements that are used to develop the Functional Requirements Document.
2.1.5	Define Data Requirements	Data Requirements are the information that is obtained for developing a detailed description of the type of data the system must use to fulfill its functional requirements. Activities that are included in the data requirements are defining the data, categorizing the data, defining data constraints, identifying input responsibilities and data collection requirements. This information is used to develop the Data Requirements Document.
2.1.6	Develop Security Plan	Outline the mechanism that will be in place to prevent unauthorized access to the project system.
2.1.7	Develop Requirements Specification Plan	Establish a common understanding between the customer and the software project of the customer's requirements that will be addressed by the software project. This document covers to the system requirements allocated to the software. The plan also covers both the technical and nontechnical (e.g., delivery dates) requirements. The agreement forms the basis for estimating, planning, performing, and tracking the software project's activities throughout the software life cycle.
C. System Design		
3.1.1	Perform Project Tracking and Control	Provides adequate visibility into actual progress so that management can take effective actions when the project's performance deviates significantly from the software plans. Project tracking and control involves tracking and reviewing the project's accomplishments and results against documented estimates, commitments, and plan, and adjusting these plans based on the actual accomplishments and results. Progress is determined by comparing the actual project size, effort, cost, and schedule to the plan when selected work products are complete at selected milestones.
3.1.2	Define/Identify System Interfaces	Identifying the system interfaces involves identifying and analyzing the external systems that will interface with the proposed system, and documenting them in the system/subsystem specification.
3.1.3	Develop System Design Specification	A document that describes the design of a system or component. The contents include system of component architecture, control logic, data structures, input/output formats, and interface descriptions.

WBS Number	Activity	Description
3.1.4	Develop subsystem Design Specifications	Developing the subsystem design specifications includes analyzing the functional and data requirements to determine which of them may be logically be processed together to form a subsystem, and developing the detailed design specification of the resulting subsystem. The product of this effort is an expanded system/subsystem specification.
3.1.5	Develop Database Design Specifications	Developing the database design specification is the effort to document the detailed design of the system databases(s). Activities include identifying the unique database names and special instructions for database use; identifying database security consideration, sensitivities, and critical issues; identifying the database management system configuration; and describing the database schema. The product of this effort is the database design specification.
3.1.6	Develop Program Design Specifications	Developing the Program Design Specifications is the effort to generate a detailed computer program specification for each primitive identified in the system and subsystem design specifications. Activities include describing the functions, logic, timing requirements, accuracy and validity of the software units; and describing the interfaces, storage requirements, input and output reports, and degree of security for the software. The product of this effort is the Program Design Specification
3.1.7	Define and Document System Components	This process outlines interfaces between components including COTS products and hardware and develop system architecture diagram.
D. Software Acquisition		
4.1.1	Perform Project Tracking and Control	Provides adequate visibility into actual progress so that management can take effective actions when the project's performance deviates significantly from the software plans. Project tracking and control involves tracking and reviewing the project's accomplishments and results against documented estimates, commitments, and plans, and adjusting these plans based on the actual accomplishments and results. Progress is determined by comparing the actual project size, effort, cost, and schedule to the plan when selected work products are completed at selected milestones.
4.1.2	Identify COTS Products	Describe the products that appear to meet requirements.

WBS Number	Activity	Description
4.1.3	Evaluate COTS Products	COTS product is evaluated throughout the entire lifetime of a system. This evaluation is done before a new system is designed, as a system is being implemented, when replacing a component or when replacing a component in an operation system. Additionally, a COTS system is also evaluated in support of policy making and product line engineering activities. The evaluation should include reviewing licensing agreements.
4.1.4	Acquire Software Products	Purchase identified COTS software.
4.1.5	Obtain Licensing Agreement	Licensing agreement provides authorization to own and operate a specific COTS product.
4.1.6	Configure/Install Software Products	Define and cost suitable service agreements to assure continuity, function and upgrades as required.
4.1.7	Perform Vendor Assessments	Determine capability and capacity of vendors to meet needed requirements
E. Hardware/Infrastructure Acquisition		
5.1.1	Perform Project Tracking and Control	Provides adequate visibility into actual progress so that management can take effective actions when the project's performance deviates significantly from the software plans. Project tracking and control involves tracking and reviewing the project's accomplishments and results against documented estimates, commitments, and plans, and adjusting these plans based on the actual accomplishments and results. Progress is determined by comparing the actual project size, effort, cost, and schedule to the plan when selected work products are completed at selected milestones.
5.1.2	Identify Hardware Products	Describe the hardware that appears to meet requirements.
5.1.3	Evaluate Hardware Products	This evaluation is done before a new system is designed, as a system is being implemented, when replacing a component or when replacing a component in an operation system. This hardware system is evaluate3d in support of policy making and product line engineering activities. Hardware products are evaluated throughout the entire lifetime of a system.
5.1.4	Acquire Hardware products	Purchase identified hardware.
5.1.5	Obtain Hardware Service Agreement	This agreement provides servicing for hardware purchased, based on specification.
5.1.6	Configure/Install Hardware Products	Define and cost a suitable service agreement to assure continuity, function and upgrades as required.

WBS Number	Activity	Description
F. New Development/Perfective Maintenance		
6.1.1	Perform Project Tracking and Control	Provides adequate visibility into actual progress so that management can take effective actions when the project's performance deviates significantly from the software plans. Project tracking and control involves tracking and reviewing the project's accomplishments and results against documented estimates, commitments, and plans, and adjusting these plans based on the actual accomplishments and results. Progress is determined by comparing the actual project's size, effort, cost, and schedule to the plan when selected work products are completed at selected milestones.
6.1.2	Develop Software Development Plan	This is a project plan for a software development project. This plan is the controlling document for managing a software project; it defines the technical and managerial process necessary to satisfy the project requirements. The plan will include the major unit of work to be completed in achieving the objective of a software project as well as specific start and end dates.
6.1.3	Develop New Software	New software is developed according to client requirements.
6.1.4	Perform Code Walkthroughs	This process is conducted to ensure that the code is free from logic errors and complies with coding standards and conventions.
6.1.5	Write Unit Test Plans	Describe the scope, approach, resources, and schedule of intended test activities: identify test items and the features to be tested, the testing tasks who will do each task, and risks requiring contingency planning.
6.1.6	Perform Unit Tests	This procedure deals with testing of individual hardware or software units or groups of related units.
6.1.7	Perform Database Administration	Provide administration to implement configuration changes.
6.1.8	Support CCB Activities	This process deals with identifying and documenting the functional and physical characteristics of a configuration item, control changes to those items, record and report change processing and implementation status, and verify compliance with specified requirements.
6.1.9	Perform Code and Document Control	Inspect and audit to verify product integrity.
6.1.10	Perform SDM Process Checks	Audit to determine process effectiveness.

WBS Number	Activity	Description
6.1.11	Support Audits	Assist in the independent examination of a work product or set of work products to assess compliance with specifications, standards, contractual agreements or other criteria.
6.1.12	Perform Network Administration	Provide any support required to maintain online system.
6.1.13	Create/Update System Documentation	The process of developing written documentation describing, defining, specifying, reporting or certifying activities, requirements, procedures or result of a software development system.
6.1.14	Configure COTS Products	This is a cost to enhance and modify existing software and hardware to meet HUD's specification.
G. System Integration and Testing		
7.1.1	Perform Project Tracking and Control	Provides adequate visibility into actual progress so that management can take effective actions when the project's performance deviates significantly from the software plans. Project tracking and control involves tracking and reviewing the project's accomplishments and results against documented estimates, commitments, and plans, and adjusting these plans based on the actual accomplishments and results. Progress is determined by comparing the actual project size, effort, cost, and schedule to the plan when selected work products are completed at selected milestones.
7.1.2	Write System Integration Test Plans	Documents the development of a system integration test plan including the tests to be run against integrated software items (program and database) and outlining the software requirements being checked as well as input data and verification criteria.
7.1.3	Perform System Integration and Test	Assures all software unit and program interfaces are present and functioning correctly as they are integrated into a total system (program and database). The activities that are included in this phase are developing and documenting test procedures, executing tests, evaluating results, and updating program documentations. Database integration and test refer to database support required to populate the physical database with live data and integrate software programs within the database.
7.1.4	Generate End of Test Report for System Integration	Document end of test report to include STAR's tested, release ID, types of testing done, identify what passed and failed.

WBS Number	Activity	Description
7.1.5	Write Requirements Verification Test (RVT) and User Acceptance Plan	Define and document the process to be followed to assure that the software development process is according to the requirements gathered and outlined the standards to be followed for acceptance testing.
7.1.6	Perform RVT and User Acceptance Testing	Determine if the software satisfies acceptance criteria. The test results are documented and traced according to the Test Plan.
7.1.7	Generate End of Test Report for RVT and User Acceptance	Documents the results of the test report for RVT user acceptance, including status and disposition of anomalies as well as any discrepancies discovered.
7.1.8	Develop Independent Validation and Verification (IV&V) Test Plan	Describe the verification and validation to be performed by an organization that is technically, managerially, and financially independent of the development and testing organizations.
7.1.9	Perform IV&V Testing	The process of observing or recording the results, and making an evaluation of some aspects of the system or component based on the verification and validation performed.
7.1.10	Generate IV&V Test Report	Describe and document IV&V activities and results, including status and disposition of anomalies in the IV&V final report
H. Installation and Deployment		
8.1.1	Perform Project Tracking and Control	Provides adequate visibility into actual progress so that management can take effective actions when the project's performance deviates significantly from the software plans. Project tracking and control involves tracking and reviewing the project's accomplishments and results against documented estimates, commitments, and plans, and adjusting these plans based on the actual accomplishments and results. Progress is determined by comparing the actual project size, effort, cost, and schedule to the plan when selected work products are completed at selected milestones.
8.1.2	Develop Installation and Conversion Plan	This document covers the period of time in the software development life cycle during which a software product is integrated into its operational environment.
8.1.3	Perform Deployment Readiness Review	Review any software processes that are ready for release to ensure that all the final product is complete and accurate.
8.1.4	Travel to Pilot and Production Site	Schedule full costs for time, transportation, accommodation, and subsistence costs need to be determined.

WBS Number	Activity	Description
8.1.5	Configure/Install at Pilot and Production Sites	“Configuring” is the process of preparing software to compile correctly on a given host, for a given target. Once the software has been configured to the given convention it is installed and used as a pilot test in the production environment.
8.1.6	Convert Data at Pilot and Production Sites	After hardware and software have been procured and modifications have been made to the physical facilities as outline in the system support plans, the procured hardware and software is installed and each component is verified individually for correct operating environment.
I. System Operation		
9.1.1	Perform Project Tracking and Control	Provides adequate visibility into actual progress so that management can take effective actions when the project’s performance deviates significantly from the software plans. Project tracking and control involves tracking and reviewing the project’s accomplishments and results against documented estimates, commitments, and plans, and adjusting these plans based on the actual accomplishments and results. Progress is determined by comparing the actual project size, effort, cost, and schedule to the plan when selected work products are completed at selected milestones.
9.1.2	Develop Training Plans	Document all information regarding the training strategy and approach in the Training Plan, in accordance with HUD SDM documentation standards. The Training Plan will include schedule of training courses, a description of each session and a list of the materials required. The training plan will also incorporate in-services coaching and support to individuals needing this assistance.
9.1.3	Prepare Training Materials	Complete all material necessary to support the scheduled training, including training booklets, online exercises, and presentation materials. Review these materials to assure compliance with the project Training Plan. Modify training materials to reflect any corrections or changes to the system that have resulted from deficiencies found during systems acceptance and pilot tests.
9.1.4	Travel to Training Site	Visit training site and prepare area for training delivery, check overhead projectors, room temperature and lighting.
9.1.5	Provide Training	Carry out the approved training sessions in accordance with the Project Training Plan. At the completion of each training session, elicit feedback from personnel to assure training objectives were met.

WBS Number	Activity	Description
9.1.6	Provide Help Desk and Customer Support	The help desk is provided for customer support and provides information to help users deal with problems related to system functionality.
9.1.7	Renew Licenses and Contractor Service Agreements	Renew the authorization given by an official or a legal authority to an individual or organization to operate or own a specific product.
9.1.8	Plan Disaster and Recovery	Contingency plan developed to document what needs to be done to mitigate problems with system functionality.
9.1.9	Purchase Spare Parts	Purchase spare parts as needed to support network and sites.
9.1.10	Support CCB Activities	This process deals with identifying and documenting the functional and physical characteristics of a configuration item, control changes to those items, record and report change processing and implementation status, and verify compliance with specified requirements.
9.1.11	Perform Code and Document Control	Perform configuration management and change control of releases and documents for the entire system.
9.1.12	Perform Library and Data Management	This process is developed to assure that system documentation and data are being controlled and will be made available to project staff when required. System documents are usually stored in a secured area. Staff members requiring use of these documents or data elements must go though a check-in and checkout process in order to maintain traceability.
J. Corrective and Adaptive Maintenance		
10.1.1	Perform Project Tracking and Control	Provides adequate visibility into actual progress so that management can take effective actions when the project's performance deviates significantly from the software plans. Project tracking and control involves tracking and reviewing the project's accomplishments and results against documented estimates, commitments, and plans, and adjusting these plans based on the actual accomplishments and results. Progress is determined by comparing the actual project size, effort, cost, and schedule to the plan when selected work products are completed at selected milestones.
10.1.2	Categorize Defects and Errors	Troubleshoot system to ascertain cause of problem and formulate solution.
10.1.3	Perform SDM Process Checks	Audit to determine process effectiveness.
10.1.4	External Audits	Provide support in response to audit findings, generate action plan and implement the plan as well as generating summary report.

WBS Number	Activity	Description
10.1.5	Perform Network Administration	Manage the development and maintenance of information in a networked environment and will provide support to end-users. Additionally develop and administer the system using relational database tools and network operating systems.
10.1.6	Update System Documentation	Update system document to assure that changes made to the system has been documented.
10.1.7	Develop Software	Code software and configure system and COTS products.
10.1.8	Perform Code Walkthroughs	This process is conducted to assure that the code is free from logic errors and complies with coding standards and conventions.
10.1.9	Write Unit Test Plans	To cover all linearly independent paths through the software components for the entire system.
10.1.10	Perform Unit Tests	This procedure deals with testing of individual hardware or software units or groups of related units.
10.1.11	Perform Database Administration	Managing planning, developing and maintaining of information in a database environment that will be available to end-users. Database administration entails development and administration of relational database tools and operating systems.
10.1.12	Support CCB Activities	This process deals with identifying and documenting the functional and physical characteristics of a configuration item, control changes to those items, record and report change processing and implementation status, and verify compliance with specified requirements.
10.1.13	Perform Code and Document Control	Inspect and audit to verify product integrity.
10.1.14	Perform Risk Management	Risk Management is the process that identifies areas of risk where corrective action is developed to reduce the risk to an acceptable level. It includes those activities required to assure efficient management of the acquisition process overall and for individual programs.
10.1.15	Generate Unit Test Results	Determine and documents the unit test results, showing traceability back to the unit test plan and software requirements.

This page intentionally left blank.

Return to Title Page

Appendix G. Software Prototyping

Return to Title Page

This page intentionally left blank.

APPENDIX G. SOFTWARE PROTOTYPING

Prototyping is the process of rapidly building a working model or mockup of a system or selected parts of a system. A prototype is capable of exhibiting the dynamics of system objects such as data entry and entry screens, output reports, system event flow, and logical data descriptions.

The primary goal of prototyping is to provide through the working model early insight into the system's characteristics. The prototype can be used to form the basis of the system requirements, or determine system feasibility, with a high level of user involvement.

The general characteristics of prototypes are the following:

- Quickly built
- Easy to change
- Normally written in 4GL or CASE
- Employ a high level of user involvement
- Are iteratively refined

Depending upon the tools used to develop the prototype, a full-scale production system may evolve from the prototype, or the production system may be implemented using a different programming language.

A prototype system may differ from a full-scale production system in several respects. Prototype systems generally have the following characteristics:

- Provide less functionality
- Are less efficient
- Are easier to change
- Are developed in a different operating environment
- Are developed using a different implementation language

The approach to be undertaken for prototyping should be determined based upon the following:

- Project scope and objectives
- Time available to prototype
- Type of hardware to be used
- Prototyping tools to be used
- Experience level of the prototyping team

Software prototyping provides the HUD organization a common framework for developing, documenting, and implementing software applications. Software prototyping techniques incorporate prototyping procedures into the traditional lifecycle described in the HUD Software Development Methodology (SDM). Development efforts involving software prototyping must supplement the lifecycle phases established by the HUD SDM with the software prototyping lifecycle stages.

This software prototyping section addresses guidelines for applications software prototyping only. Hardware prototyping and system software prototyping are not addressed. Guidelines provided include:

G.1 Prototyping Procedures

Follow the software prototyping stages (planning, development, and specification) as a supplement to the HUD SDM for all software prototyping projects. Prototyping projects that continue to full-scale development must continue to follow the HUD SDM once the final prototype has been accepted and formally specified.

Prototyping projects that require two weeks or less to develop will follow the software prototyping stages but may require less project control and documentation than those of longer duration. (See the Short-term Prototyping section of this document for more details.) Information for supplementing the SDM includes:

G.1.1 Prototyping and the HUD SDM Lifecycle

Begin all prototyping projects after the Needs Statement is approved. The Needs Statement, prepared during the Initiate Project phase, is one of the primary inputs to the prototyping project.

Once a project has been approved for prototyping by the project manager, follow software prototyping techniques until an acceptable prototype is developed. Software prototyping involves an iterative process in which each iteration results in an improved, more targeted version of the prototype. When the final prototype has been approved, re-enter the HUD SDM lifecycle and once again follow the SDM.

All software development efforts follow the HUD SDM to produce the deliverables specified in the SDM. In addition, projects using prototyping produce the deliverables specified in software prototyping.

G.1.2 General Documentation Requirements in Software Prototyping

Prototyping does not eliminate the need for documentation. Written documentation prepared throughout software prototyping describes the evolving prototype at the appropriate level of detail and provides necessary information to managers, developers, and system users.

A software prototype permits two types of documentation: a working model (including a draft User’s Manual) and traditional documentation such as data flow diagrams (DFDs), data dictionary listings, screen and report layouts, etc. Software prototyping stages are marked by specific deliverables. For details see Table G-1, HUD Prototyping Deliverables and Responsibilities, or Table G-2, HUD Short-term Prototyping Deliverables and Responsibilities.

Most documentation requirements of software prototyping reside in the areas of front-end planning; rapid analysis; and, after the prototype is accepted, the final specification task. This approach allows the developers the necessary freedom to build, demonstrate, and refine the prototype, requiring only minor updates to some of the documentation previously produced.

Table G-1. HUD Prototyping Deliverables and Responsibilities

Software Prototyping Stage	Software Prototyping Deliverable	Control/Approval Responsibility		Functional Responsibility
		User	PM	
PLANNING	Context diagram & DFDs (PP)	P	S	PM
	Project Plan (PP)	S	P	PM
	Initial Prototype	P	S	PM
	Draft User's Manual	X		U
	High-level DFDs	S	P	PM
	ERDs	S	P	PM
DEVELOPMENT	Control flow graphs	S	P	PM
	Security & control requirements	S	P	PM
	Data dictionary	S	P	PM
	Summary list of refinements (PP)	P	S	U
	Final prototype	P	S	PM
SPECIFICATIONS	Documentation package	X		PM
	Decision to implement (PP)*	P	S	U

KEY:
PM = Project Manager
U = User
P = Primary
S = Secondary
X = Exclusive
PP = To be included in the Project Plan of the project following SDM
* Concurrence between user, project manager, and IT regarding functional decision; documented in the System Decision Paper

Table G-2. HUD Short-term Prototyping Deliverables and Responsibilities

Software Prototyping Stage	Software Prototyping Deliverable	Control/Approval Responsibility		Functional Responsibility
		User	PM	
PLANNING	Context diagram & DFDs (PP)	P	S	PM
	Prototype Project Plan (PP)	S	P	PM
DEVELOPMENT	Initial prototype	P	S	PM
	High-level DFDs	X		U
SPECIFICATIONS	Documentation package	X		PM
	Decision to implement (PP)*	P	S	U

KEY:
PM = Project Manager
U = User
P = Primary
S = Secondary
X = Exclusive
PP = To be included in the Project Plan of the project following SDM
* Concurrence between user, project manager, and IT regarding functional decision; documented in the System Decision Paper

G.2 Software Prototyping Stages

The development of software using software prototyping involves three stages of activity. The initial stage of the prototyping effort, called the planning stage, involves developing a plan for carrying out the prototype. The next stage, the development stage, is when the actual prototype is built, demonstrated, and refined. The final stage, the specifications stage, is when the "as prototyped" specifications, those that will be carried back into the traditional HUD lifecycle, are determined and documented. Additional information on these stages are provided in the following sections:

G.2.1 Prototype Planning Stage

G.2.2 Prototype Development Stage

G.2.3 Specifications Stage

G.2.1 Prototype Planning Stage

Execute the prototype planning stage during the SDM Initiate Project phase. Identify the prototype initiatives and develop a Prototype Project Plan (PPP). The PPP establishes the contractual agreement between the development organization and the user organization(s). It specifies the accomplishments, by whom they will be accomplished, the time frame, the cost, and the prototype method and tools to be used, as follows:

G.2.1.1 Prototype Scope and Objectives

G.2.1.2 Prototyping Team Description

G.2.1.3 Determining and Approving Prototype Candidates

G.2.1.4 Milestones and Deliverables

G.2.1.5 Hardware, Tools, and Techniques

G.2.1.6 Estimated Prototyping Costs

G.2.1.1 Prototype Scope and Objectives

In Section 1 of the Prototype Project Plan, define the purpose of the prototyping project and establish the boundaries of the system to be prototyped. State the specific factors that determine the prototype system boundaries. These factors would typically include the user organization(s) for which the prototype is being developed; other systems that interact with the prototype and which may serve as sources or recipients of the prototype data; security requirements for hardware, software, data, and user access; creation and use of test files; and specific limitations of the prototype which may need to be addressed in the full-scale production system.

G.2.1.2 Prototyping Team Description

Identify prototyping team members by name and position. Specify the tasks and responsibilities associated with each team member, and state each individual's time commitment for each prototype task. The prototyping team description should promote better productivity because it establishes the working environment for the team. It eliminates confusion about roles and about the responsibility for deliverables of each team member.

Selection Factors

Prototype development teams should be small in order to enable effective communication and decision making. The optimal size of a prototyping team is three to five persons: one or two experienced developers (analysts/programmers) and two to three users. When development of the prototype requires the participation of more than one functional area, user groups should be formed to provide requirements to the prototyping team through the user coordinator. Large projects may require several prototyping teams, each one assigned to a separate subsystem.

The prototyping team must be knowledgeable about the HUD SDM and the related lifecycle. Developers should be trained in the techniques of prototyping and in the use of automated prototyping tools prior to the inception of the project.

All end users should be identified and represented on the prototyping team. The user representative must make a commitment to a high level of involvement during all prototyping tasks so that the project will be successful. The user coordinators of the prototyping team must be:

- Able to devote the response time necessary to define, develop, and refine the prototype
- Knowledgeable about the expected functional requirements of the system to be prototyped
- Able to accept decision-making responsibility
- Willing to consider and include security and internal control requirements
- Responsible for providing the necessary input to the prototyping team

Responsibilities

Responsibilities of the project manager for the deliverables produced during the planning stage of software prototyping include preparing the Prototype Project Plan (PPP) and obtaining milestone concurrence for the PPP. The project manager and the user representative must sign the milestone concurrence for the PPP. The project manager is ultimately responsible for determining or approving the work assignments, prototyping tools, schedules, and milestones for the prototyping project. The project team should have input to the plan and should also review it.

The prototyping team leader works under the direction of the project manager and coordinates, facilitates, and oversees the development of the prototype. The team leader ensures preparation of software prototyping deliverables and updates the central project file.

The prototype developers work under the direction of the prototyping team leader to develop and refine the prototype.

The prototype user coordinators assist in the development of the prototype and are the subject matter experts that supply information about the system and the users' needs.

The user representative is responsible for making user personnel and resources available to complete the requirements of the prototype planning stage. This responsibility also includes making available the personnel with the proper expertise (knowledge of the end-user needs) to serve on the project team.

G.2.1.3 Determining and Approving Prototype Candidates

The team will identify the portions of a system that are suitable candidates for applying the prototyping methodology. Good candidates for prototyping generally are processes that are highly interactive and relate to operational support, data management, or management information. Processes that are predominantly batch oriented or heavily algorithmic or that involve ad hoc retrieval or analysis are generally not good candidates for the prototyping methodology.

The prototyping team will develop the context diagram and high-level data flow diagrams (DFDs). High-level DFDs consist of the zero (0) level, at a minimum, and, if necessary, two to three levels of decomposition.

The context diagram and high-level DFDs must be submitted to the project manager for approval prior to the initiation of the prototype.

G.2.1.4 Milestones and Deliverables

A project management tool should be used to show work tasks, early start dates, early completion dates, late start dates, late completion dates, milestones, schedule dependencies, slack time, and critical paths. Minimal prototyping project milestones should include approval of the Prototype Project Plan, completion of the rapid analysis, demonstration of the initial prototype, demonstration of the final prototype, completion of the specification task, and the decision to implement the system.

The deliverables to be produced during software prototyping must be specified. Table G-1 and Table G-2 specify the deliverables for HUD prototyping projects. Additional deliverables may be required for specific projects and will be identified, when applicable, by the project manager.

Documentation Requirements

Prototyping project deliverables at the project planning stage of software prototyping consist of the project plan, context diagram, and high-level DFDs. The user organization and IT must concur on the project planning stage, and this concurrence must be documented in the system decision paper. The Prototype Project Plan should become part of the project file.

G.2.1.5 Hardware, Tools, and Techniques

All prototyping tools must be identified by individual tool, vendor name, and tool category (refer to Prototyping Tool Selection, Section 5.0). Describe the prototyping hardware environment and the anticipated hardware environment of the final production system.

G.2.1.6 Estimated Prototyping Costs

All costs associated with developing and testing the prototype must be estimated and specified in this section. Cost breakdowns should include personnel, hardware and software acquisition, training, site preparation, travel and per diem, and additional costs that directly contribute to the production of the prototype during software prototyping. Significant costs that are likely to be incurred during implementation of the prototype or that are required for integration into the lifecycle should also be identified.

G.2.2 Prototype Development Stage

During the development stage of software prototyping, Define, construct, and iteratively refine a working prototype until it satisfies the user's needs. The iterative process will cease once the user agrees that the prototype represents an accurate statement of their requirements.

The development stage of software prototyping requires a high level of involvement on the part of the user organization. The prototype development stage activities and responsibilities are described in the following sections:

G.2.2.1 Rapid Analysis

G.2.2.2 Build Prototype

G.2.2.3 Prototype Demonstration and Review

G.2.2.4 Prototype Refinement

G.2.2.1 Rapid Analysis

The purpose of rapid analysis in software prototyping is to learn enough about the problem to enable the construction of a good initial prototype. The prototyping team must establish a “good guess” at the requirements for the ultimate system by identifying the core set of functions and data that serve as the nucleus of the ultimate system and then capturing a significant portion of this nucleus in the initial prototype.

The typical tasks performed during rapid analysis are similar to those performed in traditional structured systems analysis, as follows:

- Review existing systems
- Interview users to establish needs
- Perform abbreviated risk analysis and determine security and control requirements
- Identify key data sources, intermediate storage devices, and interfaces
- Develop additional data flow diagrams (if necessary)

- Develop logical data models

Further details are described in Table G-3, Activities and Responsibilities for Rapid Analysis.

Table G-3. Activities and Responsibilities for Rapid Analysis

User Representative	UR	UC	Project Manager	PM	PT
1. User Coordinator may convene with user group(s) to obtain input for identification of specific needs.		X	1. May attend user group(s) to obtain direct insight of specific needs.	X	X
2. Assist in preparation of working documents and provide input to describe prototype requirements.		X	2. Prepare working documents for initial prototype: • High-level DFDs • Control flow graphs • Security & control requirements • Partial data dictionary		X
3. Review working documents to ensure technical adequacy and consistency with project scope and objectives.		X	3. Review working documents to ensure technical adequacy and consistency with project scope and objectives.		X
			4. Develop logical data model entity relationship diagram.		X
			5. Update Prototype Project Plan if necessary.*	X	
6. User Representative and Project Manager agree that the scope of the initial prototype is acceptable.					

KEY:
PM = Project Manager
UR = User Representative
UC = User Coordinator(s)
PT = Prototyping Team Leader and Prototype Developer(s)
* Not required for short-term prototyping

Documentation to be developed during rapid analysis describes the partial requirements for the system and includes additional data flow diagrams, entity relationship diagrams, control flow graphs, security and control requirements, and a partial data dictionary.

The prototyping team leader must review the documentation produced during rapid analysis before initial prototype development begins in order to ensure that a sufficient set of functions have been identified for the nucleus of the initial prototype.

G.2.2.2 Build Prototype

The initial prototype should demonstrate the prototyping team's understanding of the core system requirements. The initial prototype must be a working entity that demonstrates, at a minimum, menus and screens and the user/system interface.

The initial prototype should emphasize functional content rather than formatting aesthetics or performance optimization. The prototype should include enough of the core functions to ensure ongoing user interest.

To be effective and retain the user's interest, the initial prototype should be delivered as soon as possible. The scope of the prototype should be such that delivery of the initial prototype of a large system is feasible within a two- to three-month time frame, assuming that the prototyping team is trained and is assigned to the project full time. Some prototypes may require only two to three days.

A top-down approach should be used to develop the prototype. The approach will vary depending on the prototyping method selected and the prototyping tools being used, but it generally will include the following activities:

- Develop and test menu screen.
- Develop and test data entry screens.
- Develop reports.
- Implement a test database.
- Develop and test functions.
- Test the prototype system.
- Implement refinements.
- Prepare a draft User's Manual.

Further details are described in Table G-4, Activities and Responsibilities for Build Prototype.

The prototyping team must prepare a draft User's Manual as an accompanying document to the working prototype. A sample document outline is presented in Table G-5, User's Manual Outline.

Table G-4. Activities and Responsibilities for Build Prototype

User Representative	UR	UC	Project Manager	PM	PT
1. Participate in construction of prototype menus, screens, reports, and functions.		X	1. Develop prototype menus, screens, reports, and functions.		X
2. Provide test data and participate in prototype testing.		X	2. Develop prototype databases and participate in testing the prototype.		X
3. Specify prototype refinements.	X	X	3. Develop prototype refinements.		X
4. Prepare draft User's Manual and demonstration materials.		X	4. Review draft User's Manual and demonstration material. Provide assistance if required.*	X	
5. User Representative and Project Manager agree upon readiness for the initial prototype demonstration. Concurrence on the initial prototype must be documented in the System Decision Paper.					

KEY:
PM = Project Manager
UR = User Representative
UC = User Coordinator(s)
PT = Prototyping Team Leader and Prototype Developer(s)
***** Not required for short-term prototyping

Return to Title Page

Table G-5. User's Manual Outline

I. Title Page
- A. Name of application
  - 1. Application description title
  - 2. Functional office
- B. Completion date of project
- C. Developing site
- D. Local contact
  - 1. Name
  - 2. Function
  - 3. Telephone number

II. Table of contents (self explanatory)

III. Introduction
- A. Application abstract (description of application and processing [equipment] environment)

IV. Operating Procedures
- A. Equipment orientation for users
  - 1. On/off procedures
  - 2. Log on (insert diskette procedures, setup procedures [if appropriate])
  - 3. Getting started procedures
- B. General instructions
  - 1. Entering data
  - 2. Changing data
  - 3. Deleting data
  - 4. Generating output products
  - 5. Query
  - 6. Other
- C. Menus
  - 1. Layout
  - 2. Explanation of menu selections
- D. Step-by-step instructions
  - 1. Input screen(s)
  - 2. Explanation of key-in data (input item, data elements, transcription procedures)

V. Output
- A. Screen layouts
- B. Hard copy reports

VI. Troubleshooting section

G.2.2.3 Prototype Demonstration and Review

The purpose of prototype demonstration and review is to develop new and revised requirements, or approve an enhancement, as a result of having all necessary people observe, critique, and experience the model. The demonstration format depends on the prototype model. Prototypes can evolve either continuously or in a succession of discrete releases with step-by-step prototyping.

Continuous Evolution

Continuous evolution involves reviewers (users) working regularly with the prototype developers, examining each enhancement when it is working.

Discrete Releases

For discrete releases, create a list of enhancements and set a target date for demonstrations and when the next version will be available for review. A prototype demonstration will be held after development of the initial prototype and subsequent to each set of refinements made thereafter by the prototyping team.

Prototype demonstration attendees should consist of a representative cross-section of the prototype system's ultimate user community. For prototype systems that have impact on a large number of functional areas, several demonstrations may need to be presented at the end of each refinement cycle in order to accommodate the larger community.

Demonstration agendas, the prototype User’s Manual, and other necessary presentation materials should be distributed to attendees in advance of each demonstration.

Further details are described in Table G-6, Activities and Responsibilities for Prototype Demonstration.

Table G-6. Activities and Responsibilities for Prototype Demonstration

User Representative	UR	UC	Project Manager	PM	PT
1. Participate in prototype demonstrations.	X	X	1. Participate in prototype demonstrations.		X
2. Prepare summary list of refinements subsequent to each prototype demonstration.	X	X			
3. Evaluate adequacy of prototype and determine if project should continue.	X	X			
			4. Update Prototype Project Plan if required.*	X	
5. User Representative and Project Manager agree upon summary list of refinements. Concurrence is required for the Summary List of Refinements at each demonstration and must be documented in the System Decision Paper.					

KEY:
PM = Project Manager
UR = User Representative
UC = User Coordinator(s)
PT = Prototyping Team Leader and Prototype Developer(s)
***** Not required for short-term prototyping

G.2.2.4 Prototype Refinement

The purpose of prototype refinement is to implement changes that are requested as a result of the prototype demonstrations, ultimately bringing the prototype in line with user expectations. Prototype refinement is performed after each demonstration until final acceptance is obtained, signifying that the prototype reflects the user's requirements for the production system.

The activities to be performed during the prototype refinement task are as follows:

- Implement the requested changes and modifications.
- Thoroughly test all changes and modifications.
- Modify the working documentation and User’s Manual.
- Conduct a prototyping team walkthrough to assess readiness for a demonstration of the revised prototype.

Modification of working documentation may require performing any number of the following tasks:

- Revise the data flow diagrams and the logical data model.

Return to Title Page

- Revise the data dictionary components.
- Revise screen, report, and function specifications.
- Revise the User's Manual.
- Document problems and limitations discovered during the refinement process.

A maximum of two refinement cycles subsequent to development and demonstration of the initial prototype is suggested in order to control costs, maintain project momentum, and to facilitate adherence to project schedules.

Documentation Requirements

Expand, update, and refine the high-level data flow diagrams (DFDs), entity relationship diagrams (ERDs), control flow graphs, security and control requirements, partial data dictionary, and User's Manual for each prototype demonstration. Prepare a summary list of refinements at each refinement cycle that becomes part of the project file.

Deliverables

After the initial prototype is demonstrated, a milestone concurrence is signed. A draft User's Manual must be delivered with the initial prototype and with each successive refinement that is demonstrated.

A summary list of refinements is required after each prototype demonstration.

The final prototype, User's Manual, context diagram, high-level DFDs, ERDs, control flow graphs, security and control requirements, partial data dictionary, and signed milestone concurrence are delivered to the full-scale system developers by the project manager upon acceptance of the prototype by the user representative.

Further details are described in Table G-7, Activities and Responsibilities for Prototype Refinement.

Table G-7. Activities and Responsibilities for Prototype Refinement

User Representative	UR	UC	Project Manager	PM	PT
1. Assist in developing and testing refinements.		X	1. Develop refinements.		X
2. Assist in updating working documents.		X	2. Update working documents.		X
3. Update User's Manual and prepare demonstration materials.*		X	3. Review updated User's Manual and demonstration materials.*		X
4. User Representative and Project Manager agree upon completion of refinement tasks and readiness for demonstrating revised prototype. Concurrence is required subsequent to the final prototype demonstration and must be documented in the System Decision Paper.					

KEY:
PM = Project Manager
UR = User Representative
UC = User Coordinator(s)
PT = Prototyping Team Leader and Prototype Developer(s)
* Not required for short-term prototyping

G.2.3 Prototype Specifications Stage

The purpose of the software prototyping specifications stage is to decide the future direction of the prototype project and to develop a formal set of “as prototyped” specifications, to include only the components of the prototype, that will serve as baseline documentation for the remainder of the lifecycle. Further details are provided in Table G-8, Specifications Stage Activities and Responsibilities.

Based on the results of the prototype development and refinement efforts, project development may proceed in one of three directions:

- Development of a production system may begin.
- Production system development and implementation may be postponed.
- Plans for development of a production system may be canceled.

The direction of a prototype may result in one of the following:

G.2.3.1 Re-entering the HUD SDM Lifecycle

G.2.3.2 Termination of software prototyping

Table G-8. Specifications Stage Activities and Responsibilities

User Representative	UR	UC	Project Manager	PM	PT
1. Prepare list of requirements for production system that were not addressed in the prototype.*		X	1. Review list of requirements.1		X
2. Assist in preparation of the Documentation Package.		X	2. Prepare the Documentation Package.		X
3. Determine Project direction.	X				
			4. Determine documentation requirements for merging into HUD's lifecycle.	X	
5. User Representative and Project Manager agree upon project direction. A decision to implement development of production system and a signed concurrence is required for projects that will re-enter the HUD lifecycle. This concurrence should be documented in the System Decision Paper.*					

KEY:
PM = Project Manager
UR = User Representative
UC = User Coordinator(s)
PT = Prototyping Team Leader and Prototype Developer(s)
* Not required for short-term prototyping

G.2.3.1 Re-entering the HUD SDM Lifecycle

Prototypes, including short-term prototypes, re-enter the project lifecycle at the Define System phase. If the prototype is the vehicle for the final product, the prototype may merge into the lifecycle at the Design System phase.

G.2.3.2 Termination of Software Prototyping

Once the production system is installed and operational and any initial problems have been resolved, conduct a post-implementation review as described in the HUD SDM Operate System phase. This review should contain a section for assessing the value of incorporating the Software Prototyping Methodology into the lifecycle.

Objectives of the Software Prototyping Post-Implementation Review

The software prototyping post-implementation review should reveal the effectiveness of prototyping and should measure the impact on the production system of having applied software prototyping.

Software Prototyping Assessments

Ascertain the effectiveness of the prototype methodology by interviewing the user and determining the answer to two questions:

- Was the system that was delivered the one the user wanted?
- Is the user satisfied with the system?

The impact of software prototyping on the production system can be measured in terms of cost avoidance in the Operate System phase of the lifecycle. Use the summary list of refinements in the project plan, in addition to records maintained by the maintenance personnel, to provide the following assessment information:

- Were additional requirements discovered during the prototype demonstrations which would have otherwise resulted in costly modifications to the system when it was placed in production?
- Were modifications made to the prototype during the refinement task of software prototyping because of a misunderstanding of the user's original requirements?
- Have there been any new features or functions added to the system while in production?
- Have there been any changes to the input or output of the system while in production?

Software Prototyping Post-Implementation Review Uses

Forward the review findings to the management official(s) responsible for the development and implementation of the system. The findings should be used by project management to determine the impact of using software prototyping.

Place the review findings and impact statement in a historical file as an audit trail.

Documentation Requirements

Prepare a documentation package for all prototyping projects. The documentation package consolidates all of the prototype specifications and information necessary to prepare the HUD lifecycle deliverable documentation. Refer to Table G-8, Specifications Stage Activities and Responsibilities, for additional information.

The documentation package provides much of the documentation requirements of the HUD SDM. The extent of the prototyping effort will determine the contents of the package. At a minimum, the package will consist of the following items:

- Context diagram and DFDs
- Entity relationship diagrams (ERDs)
- Control flow graphs
- Security and control requirements
- Data dictionary
- Screen layouts
- Report layouts
- Menus
- Logical database model
- Physical database structure
- Draft User’s Manual
- Requirements not addressed by the prototype

A significant portion of the information contained in the documentation package may be supplied by the tools used during the prototyping process.

Before the prototype development process may re-enter any phase of the lifecycle, all prior phase deliverables of the HUD SDM which were not manually produced, or produced by the automated prototyping tool, must be developed.

Deliverables

The documentation package is prepared for all prototyping projects regardless of whether a production system will be developed.

A decision to develop or to not develop a production system is required for all projects. Record the decision in the System Decision Paper.

G.3 Short-term Prototyping

Prototyping projects requiring two weeks' development time or less may be categorized as short term and qualify for the exclusions defined in this section.

The project manager is ultimately responsible for determining and specifying adequate project controls and documentation requirements for projects in this category. However, prototyping projects that continue to full-scale development must continue to follow the HUD SDM once the final prototype has been accepted and the prototype documentation package has been completed.

When the Define System phase of the lifecycle is re-entered, the production system developers will use the system concept and user requirements demonstrated by the prototype as the basis for completion of the requirements of the HUD SDM.

The outline that follows contains recommended maximum exclusions of project controls and documentation during the software prototyping task. Refer to Table G-2, HUD Short-term Prototyping Deliverables and Responsibilities, for additional information.

G.3.1 Exclusions for the prototype planning stage

G.3.2 Exclusions for the prototype development stage

G.3.3 Exclusions for the specifications stage

G.3.1 Exclusions for the Prototype Planning Stage

- Milestones and deliverables
- Hardware, tools, and techniques
- Milestone concurrence for the Prototype Project Plan

G.3.2 Exclusions for the Prototype Development Stage

- Entity relationship diagrams
- Control flow graphs
- Security and control requirements
- Partial data dictionary

- Review of Rapid Analysis documentation
- Draft User's Manual and demonstration agendas
- Modification of working documentation
- Summary list of refinements
- Concurrence on delivery of the initial prototype

G.3.3 Exclusions for the Specifications Stage

- Entity relationship diagrams
- Control flow graphs
- Security and controls requirements
- Data dictionary
- Screen layouts
- Report layouts
- Menus
- Logical database model
- Physical database structure
- Draft User's Manual
- Requirements not addressed by the prototype

G.4 Prototyping Tool Selection

The use of automated prototyping tools is an important factor in making prototype development successful. Automated prototyping tools allow programs, data, and database structures to be constructed and modified rapidly and easily.

Most prototyping tools can be classified as the following:

- Screen generators
- Report generators
- Relational database management systems (RDBMS)
- Fourth generation languages (4GLs)
- Spreadsheets

Prototyping tools selected for a project should be based on the project scope, objectives, type of application, and HUD standards.

The Prototyping Tool Evaluation Checklist presented in Table G-9 provides a series of questions that can be used in the evaluation and selection of prototyping tools. This list of questions may or may not apply to all prototyping tool categories and is not all-inclusive. Each of these factors may be weighted to provide a total score to assist in evaluating the applicability of each prototype tool that is being considered for the application. The weighting factors should be assigned relative to features that are necessary (1), are highly desirable (2), or are moderately desirable (3). The total score should be evaluated further on the basis of the prototyping tool having the greater number of necessary features.

Table G-9. Prototyping Tool Evaluation Checklist (1 of 4)

CRITERIA	WEIGHTS		
1. Screen Features	**1**	**2**	**3**
a. Does it have screen generation capabilities?			
b. Is it easy to design and change screen formats?			
c. Does it use color?			
d. Does it provide windowing capabilities?			
2. Operating Features	**1**	**2**	**3**
a. Does it provide networking capabilities?			
b. Does it allow real-time processing?			
c. Is response time acceptable?			
d. Does it allow multi-user access?			
e. Can it operate within the current hardware environment?			
f. Is its impact on computer resources acceptable?			
g. Can it interface with the operating system?			
h. Are the main memory requirements acceptable?			
i. Are the hard disk storage requirements acceptable?			
j. Are the peripheral device requirements acceptable?			
3. User Interface Features	**1**	**2**	**3**
a. Does it have help facilities for all functions?			
b. Does it have program/data/text editing capabilities?			
c. Does it support menus?			
d. Are both novice and expert modes available?			
e. Does it have defaults to help novice users?			
f. Can experts override the defaults?			
g. Does it eliminate the need for the operator to remember mnemonics, formats, or fixed sequences?			
h. Is it self-teaching, with computer-aided instruction?			
i. Is it easy to install? Can it be installed without the assistance of systems programmers?			
j. Does it have simple, obvious-to-use sign-on procedures?			
k. Does it provide clear, self-explanatory error messages?			

Table G-9. Prototyping Tool Evaluation Checklist (2 of 4)

CRITERIA	WEIGHTS		
4. Language Features	**1**	**2**	**3**
a. Is there a non-procedural user language?			
b. Is the language portable to other operating environments?			
c. Is the developed code portable to other operating environments?			
d. Are the language and developed code portable between micro-, mini-, and mainframe computer environments?			
e. Can it interface with other programs written in a different language?			
f. Can it interface with existing databases?			
g. Can it interface with existing data dictionaries?			
h. Will it process existing files, record types and sizes?			
i. Are the table size restrictions and search methods adequate?			
j. Does it have an Artificial Intelligence or expert system interface?			
k. Does it have an interpretive mode for development and a compile mode for optimized operation?			
l. Is there a procedural, command, or programming language?			
m. Can users search with Boolean logic and logical comparison operators (and, or, $<$, $>$, $=$, $\neq$)?			
n. Does it produce source code?			
o. If source code is produced, is the programming language acceptable?			
p. Can the system process parameter lists at runtime?			
q. Are the language and syntax the same for different components (e.g., query, report generator)?			
r. Does it utilize SQL?			
s. Can the language support mathematical functions? Statistical functions?			
t. Are there intelligent default functions if details are unspecified?			
u. Can it also be used as a development language?			
v. Does the language syntax lend itself easily to maintenance?			
w. Can it operate at different levels of verbosity in its dialogue?			
x. Can the user change the sequence of computations?			

Return to Title Page

Table G-9. Prototyping Tool Evaluation Checklist (3 of 4)

CRITERIA	WEIGHTS		
5. Data Management Features	**1**	**2**	**3**
a. Is there a data dictionary?			
b. Is there an active data dictionary?			
c. Does the system allow text or variable-length records?			
d. Will the database accept foreign files and create files for foreign systems?			
e. Can the user add or delete fields, columns, or records in the database?			
f. Can the user perform sorting functions?			
g. Is the maximum allowable size adequate for files, records, fields, fields per record, characters per field?			
h. Can data files be reorganized without re-entering or loading files?			
i. Can data be entered in batches (not record by record)?			
j. Can the system process multidimensional arrays or matrices?			
k. Does the system allow and control concurrent access to the database?			
l. Does it support an RDBMS?			
m. Can new data be added while other users are accessing its database?			
n. Can new secondary indices be added for any field? At any time?			
o. Can its data dictionary be linked to a major DBMS dictionary?			
p. Can the data dictionary use aliases?			
q. Is backup and recovery of the database provided?			
6. Report Writing Features	**1**	**2**	**3**
a. Is there a report generator facility?			
b. Are there defaults for simple reports?			
c. Are there capabilities for complex reports?			
d. Does it process subtotals, column totals, row totals, percent calculation across columns, percent calculation down columns, accumulations?			
e. Does it provide footnotes, more than one layer of headings, automatic pagination, automatic spacing, processing of multiple reports in one pass?			
f. Does it allow user-defined reports by data item, zero suppression, floating dollar signs, placing commas in numeric fields, user-specified spacing, user-specified page breaks, table lookup, printing odd-sized paper, row titles?			
g. Does it allow a number of fonts to be specified and accessed?			

Table G-9. Prototyping Tool Evaluation Checklist (4 of 4)

CRITERIA	WEIGHTS		
7. Graphics Features	**1**	**2**	**3**
a. Is there a graphics generator? b. Are there facilities for producing line graphs, bar charts, pie charts, scatter plots?			
8. Security Features	**1**	**2**	**3**
a. Is security based on password protection? b. Are there additional security features such as scrambled passwords, encryption, audit trails? c. Is there password protection at the system level, database level, file level, record level, data item level? d. Does the prototyping tool track and report unauthorized attempts to enter the system? e. Are logs of usage maintained at the system level? f. Are there levels of security based on command type? g. Does the prototyping tool need to interface with existing security software?			
9. Vendor Support Features	**1**	**2**	**3**
a. Does the vendor provide training, maintenance, hot-line support, on-line documentation, reference manuals? b. Is the user documentation easy to follow? c. Is there an installation manual, summary of commands? d. Will the vendor customize the system? e. Will the most current documentation or manual releases be automatically distributed? f. Is some training provided with the purchase price? g. Is the total purchase price acceptable?			

G.5 Data Administration/Data Dictionary

The data dictionary is a central repository of data that describes data. It consists of an inventory of data types and provides names, data structures, and information regarding data usage.

The data dictionary system is a computer software system that provides for recording, storing, and processing information about an organization's data and data processing entities.

All entries to the data dictionary should be consistent within that dictionary and the production dictionary. The format may be developed manually, or tools such as a word processor or an automated data dictionary may be used. Some automated tools used in the prototype development process may also provide an active data dictionary capability. Use an active data dictionary to save time producing this deliverable.

Keep the data dictionary updated throughout software prototyping. It should become part of the documentation package of the specifications stage.

Data developed or collected during software prototyping should be named according to HUD data naming standards.

G.6 Controls

The *Model Framework for Management Control Over Automated Information Systems* (OMB January 1988) states: "At a technical level, there are at least three efforts that management should make to ensure that a hastily assembled prototype does not become operational without appropriate controls." HUD is responsible for safeguarding the integrity of the data it maintains on computerized files, as well as for complying with the provisions of the Privacy Act of 1974 and the Freedom of Information Act. Live data should not be used during prototyping. Prototyping controls are discussed in the following subsections:

G.6.1 System Controls

System controls are typically required for projects that are developed using the traditional HUD system development lifecycle. Depending on the scope and objectives of a prototyping application project, it may be necessary to establish a system control point (point of contact) at the Regional or Field Office level.

- During compressed system Initiate Project, Define, Design, Build, and Evaluate phases, begin concurrent documentation of control objectives and tentative control techniques.
- Include controls that require programming early in the prototype whenever possible.
- For controls that are to be developed as prototype modifications, ensure that the controls are actually developed and implemented at some point before implementation.

Reference the Federal Manager's Financial Integrity Act and OMB Circular A-123 for additional system controls information.

G.6.2 Security Controls

Security controls must be established for all projects within HUD. These controls include the physical security of the ADP equipment as well as the security measures necessary for safeguarding the computer programs and data files used in the prototyping project. Refer to HUD Security Directive 2400.23 for information regarding the establishment of security controls. Additionally, refer to OMB Circular A-130 (Appendix III), Paperwork Reduction Act; P.L. 100-235, Computer Security Act of 1987; OMB, January 1988 (PCMI - PCIE), Model Framework for Management Control Over Automated Information Systems.

This page intentionally left blank.

Return to Title Page

Acronyms

This page intentionally left blank.

ACRONYMS

ACWP	actual cost of work performed
ADP	automated data processing
ARN	Advanced Requirements Notice
BCWP	budgeted cost of work performed
BCWS	budgeted cost of work scheduled
CASE	Computer-Aided System Engineering
CASH	Comprehensive Administration Services at HUD
CBT	computer-based training
CCB	Configuration Control Board
CI	configuration item
CM	Configuration Management
CMM	Capability Maturity Model
CMP	CM Plan
COTS	commercial off-the-shelf
CPU	central processing unit
CSA	Configuration Status Accounting
CSOMG	Computer Services, Operations, & Maintenance Group
CV	cost variance
DASD	direct access storage device
DBA	Database Administrator
DBMS	Database Management System
DFMSSIP	Department Financial Management Systems Strategic Integration Plan
DTAD	Development Technology Assessment Division
EOR	Evaluation of Request
FCA	Functional Configuration Audit
GB	gigabyte

Return to Title Page

GOTS	government off-the-shelf
HIIPS	HUD Integrated Information Processing Services
HUD	U.S. Department of Housing and Urban Development
ID	identification
IG	Inspector General
ISP	Information Strategy Plan
IT	Information Technology
JAD	Joint Application Development
kB	kilobyte
LAN	local area network
MB	megabyte
MTBF	mean time between failures
MTTF	mean time to failure
NOS	Network Operating System
NS	Needs Statement
NS/ARN	Needs Statement/Advanced Requirements Notice
ODC	other direct costs
OIT	Office of Information Technology
OMB	Office of Management and Budget
PC	personal computer
PTAR	Problem Tracking Report
PTARS	Problem Tracking and Reporting System(s)
QA	Quality Assurance
RAD	Rapid Application Development
RAM	random access memory
RDBMS	Relational Database Management System
RM	Requirements Management
SCI	software configuration item

SCM	software configuration management
SDM	System Development Methodology
SDP	System Decision Paper
SEO&PMD	Systems Engineering, Oversight & Performance Management Division
SEI	Software Engineering Institute
SPP	software project planning
SPTO	software project tracking and oversight
SQA	software quality assurance
SQL	Structured Query Language
STAR	Service Ticket Action Resolution
STARS	Service Ticket Action Resolution System
SV	scheduled variance
TBD	to be determined
TIB	Technology Investment Board
VV&T	validation, verification, and test
WBS	work breakdown structure

Return to Title Page

This page intentionally left blank.

References

This page intentionally left blank.

REFERENCES

1840.1 HUD Handbook 1840.1. Departmental Management Control Program.

Handbook prescribes the policy for the Department's Management Control Program which was established to help ensure that effective systems of internal control are maintained, as required by law.

2229.1 HUD Handbook 2229.1. Records Disposition Scheduling for Automated Systems.

Handbook provides instructions for ADP system sponsors, assisted by the Office of Information Technology (IT), for establishing retention periods for mainframe computer files. Systems operated by outside contractors are also subject to these procedures.

2400.1 HUD Handbook 2400.1. Information Resources Management (IRM) Policies.

This handbook: (1) establishes IRM policies based on OMB Circular no. A-130, Management of Federal Information Resources; (2) establishes and Information Resources Management Planning Board (IRMPB); (3) provides the role of the senior official and IRM responsibilities in the Office of Administration.

2400.15 HUD Handbook 2400.15. HUD SDM Documentation Standards.

Handbook sets forth guidelines for the development and maintenance of documentation for HUD's ADP systems.

2400.18 HUD Handbook 2400.18. User's Guide to HUD ADP Standards.

Handbook provides guidance in the use of HUD's computing services. Supplements must be used in conjunction with either the User's guide to SPERRY Standards, or the Users Guide to IBM Standards, as applicable.

2400.23 HUD Handbook 2400.23. ADP Security.

Handbook outlines the policies and procedures for the development, implementation, and maintenance of HUD's ADP security program.

2400.24 HUD Handbook 2400.24. ADP Security Program.

Handbook establishes the ADP security program for HUD. It describes HUD security policies and organizational responsibilities for maintaining the integrity, availability, and confidentiality of HUD's information systems.

ADP Environment Guide. May, 1993.

Documents HUD's technical platform including, but not limited to hardware, support software, and communications for all information processing.

Benefit/Cost Analysis, Volume I - Methodology. September, 1995

This document includes a brief conceptual overview of economic analysis concepts, and goes on to define, describe and discuss each of the component parts of a benefit/cost analyst.

Benefit/Cost Analysis, Volume II - Workbook. September, 1995

This workbook is intended as a “how to” guide in performing a benefit/cost analysis. It contains instructions, checklists, and tables which should enable analysts to organize and perform a benefit/cost analysis.

CASE Usage Guide. September, 1991.

This guide documents Information Engineering Methodology (IEM) and Computer Aided Software Engineering (CASE) technology transfer; the IEM life cycle phases and deliverables, including mapping IEM deliverables to SDM deliverables; and Texas Instruments Information Engineering Facility (IEF) tool. The document also addresses HUD standards for information engineering and the IEF encyclopedia; and guidance for system construction using the IEF.

Configuration Management Guide. To be developed.

The governing document that provides guidelines and procedures for establishing and operating a configuration management function for a system development effort.

Data Administration Standards and Procedures. December, 1994.

This document provides information on data administration fundamentals; data usage standards; and data administration deliverables.

HUDNET File Transfer Procedures. February, 1991.

Documents the procedures for transferring large data files across different hardware platforms via HUDNET.

Information Resources Management (IRM) Plan. Appendix: IRM Planning and the Systems Development Cycle. May, 1992.

Documents the interrelationship between IRM planning and the SDM. The appendix describes IRM budget formulation; and outlines the responsibilities of personnel involved with obtaining approval of a software development project in regards to IRM activities.

LAN Administrators Guide. April, 1994

The governing document that provides standards to assure consistent configurations and operations across all HUD LANs. Designed to facilitate long term support and maintenance of the large number of HUD LANs and improve communications between LAN Administrators and HUD Headquarters Support Staff.

LAN Developers Guide. September, 1992.

The document that outlines the standards and procedures for developing application systems for HUD's Local Area Network (LAN) platform. Details are provided to software developers on the HUD LAN software and hardware environment.

LAN Systems: Application Software Release Procedures. March, 1996

Model Framework for Management Control Over Automated Information Systems. January, 1988.

Prepared jointly by the President's Council on Management Improvement and the President's Council on Integrity and Efficiency, January 1988. This report synthesizes the various directives which contain guidance on how to protect Automated Information Systems.

Platinum Report Facility (PRF) Policy and Procedures. April, 1992.

Documents the procedures for running the PRF so that no machine performance problems will be encountered. Procedures include information on accessing the PRF; submitting batch reports; testing reports; and running on-line reports.

Procedures for Logic Information Network Compiler (LINC II) Usage. August, 1992.

Provides procedures for use by software development staff utilizing the LINC II CASE tool. Procedures include registering a LINC project; resolution of system development problems; and model uploads.

Project Management Guide. To be developed.

The governing document which provides guidelines and procedures for performing project management for a software development effort.

Quality Assurance Guide. To be developed.

The governing document which provides guidelines and procedures for establishing and operating a quality assurance function for a system development effort.

Standard Release Procedures. August, 1992.

Provides HUD with a uniform set of procedures for releasing software into the production environment, includes release procedures for all hardware platforms. This guide Incorporates the LAN Release Procedures.

Software Engineering Institute (SEI) Capability Maturity Model (CMM), Version 1.1. February, 1993.

This document guides software organizations in the control of their processes for developing and maintaining software, and steadily improving their organization-wide software processes to enable continuous gains in software process capability.

Key Practices of the Capability Maturity Model, Version1.1. February, 1993.

This document provides descriptions of key process areas required to reach Software Engineering Institute (SEI) Capability Maturity Model (CMM) levels.

Return to Title Page

This page intentionally left blank.

Return to Title Page

GLOSSARY

This page intentionally left blank

Glossary of Terms

This Glossary of Terms is intended to provide an authoritative set of terms to be consistently used throughout HUD. It will grow as more terms are identified and defined.

Word/Term	Definition	Reference
Adaptive Maintenance	Modifications that are mandated in response to laws or regulations that govern HUD programs; to hardware or software changes, and/or upgrades within HUD's technical architecture.	SDM Appendix D: System Maintenance, p. D-4
Advanced Requirements Notice (ARN)	Document(s) describing the deficiency and justifying the exploration of alternative solutions.	SDM Section 1: Initiate Project, p. 3
Audit	An independent examination of a work product or set of work products to assess compliance with specifications, standards, contractual agreements, or other criteria.	IEEE Std 610.12-1990
Baseline	A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures. A document or a set of such documents formally designated and fixed at a specific time during the life cycle of a specific item. Any agreement or result designated and fixed at a given time, from which changes require justification and approval.	IEEE Std 610.12-1990
Benchmark	A standard against which measurements or comparisons can be made. A procedure, problem, or test that can be used to compare systems or components to each other or to a standard.	IEEE Std 610.12-1990
Benefit/Cost Analysis	An inspection of all economic (tangible and intangible) benefits and costs, including development and operation, one-time, and recurring costs, are calculated in current dollar values. Benefits that cannot be quantified in dollar amounts are identified. Each alternative is then evaluated based on the net economic benefits of developing and operating the system, as calculated during the analysis. This evaluation is documented in the benefit/cost analysis, including the analysis methodology, the alternatives with their associated benefits and costs, and a recommendation as to the best development alternative.	SDM Section 1: Initiate Project, p. 3
Build System Phase	Involves developing and testing computer programs in response to approved specifications and preparing documentation needed for the ongoing operation and maintenance of the system.	SDM Introduction, p. 5

Return to Title Page

Word/Term	Definition	Reference
Change Control Board (CCB)	The decision-making body for each program area project that is the control mechanism for the program office that has requested the need for which the project has been initiated. The CCB consists of program officials, system users, external stakeholders, and IT representatives, all appointed by the project sponsor. The CCB evaluates the scope, applicability, and effect of each requested change, focusing on the items that affect cost, schedules, or compliance with technical requirements, and providing approval and disapproval based on defined strategic initiatives, program business objectives, and budgetary parameters. The CCB has the authority to establish baselines, initiate or change software, accept testing results, and approve the release of software into production.	SDM Section 1: Initiate Project, p. 4
Change Control System	A collection of formal, documented procedures that defines the steps by which official project documents may be changed, including paperwork, tracking systems, and approval levels necessary for authorizing changes.	IEEE Std 1490-1998, IEEE Guide-Adoption of PMI Standard-A Guide to the Project Management Body of Knowledge, Section 4.3.2.1
Computer-aided Software Engineering (CASE) tool	A software tool that aids in software engineering activities, including, but not limited to, requirements analysis and tracing, software design, code production, testing document generation, quality assurance, configuration management, and project management.	IEEE Std 1348-1995
Configuration Audits	Conducted at the completion of system integration testing and before the software is migrated to the production environment.	SDM Appendix B: CM Guidelines, p. B-10
Configuration Control	Depends on placing products under control at the right time and on establishing mechanisms for controlling changes to the products.	SDM Appendix B: CM Guidelines, p. B-7
Configuration Identification	The process of identifying items to be placed under configuration control.	SDM Appendix B: CM Guidelines, p. B-4
Configuration Item (CI)	An integrated set of system components and may be composed of any combination of components either developed by the project or purchased for the project.	SDM Appendix B: CM Guidelines, p. B-4
Configuration Management (CM)	Continual, consistent documentation of the development and evolution of the system, to ensure that all milestones are met, key decisions are recorded, the system can be accurately described, and a consensus is reached on what products are required for delivery. CM must maintain a controlled library of project products, software, hardware, and documentation and provide a process for the consideration and disposition of requested modifications to the system.	SDM Introduction, p. 7
Configuration Management (or Project) Library	A central library for a project where baselined products (CIs) that have been completed and approved, according to HUD procedures, are stored, updated versions are kept, and access is controlled by the CM.	SDM Section 5: Evaluate System, p. 21

Return to Title Page

Word/Term	Definition	Reference
Configuration Management Plan (CMP)	The CMP is where the project's CM program is documented and is either a subsection of the Project Plan or a standalone document. The CMP is used to as the basis for performing CM and defining CM activities, responsibilities assigned to CM, and required resources needed by CM. The CMP is prepared in parallel with the project planning and requires project and IT management approval.	SDM Appendix B: CM Guidelines, p. B-3
Configuration Management Standards and Procedures	The specific processing steps required to effectively implement CM in different environments, namely, mainframe, LAN/PC, and existing legacy systems. These procedures document the methods for performing CM activities and take into account the system's environmental elements, such as distributed systems with remote sites, multiple versus single system installations, revisions, centralized software control, multiple hardware configurations, and organizational interfaces and responsibilities.	SDM Appendix B: CM Guidelines, p. B-10
Configuration Status Accounting (CSA)	An element of CM that focuses on recording and monitoring changes to controlled system configurations and maintaining a controlled documentation library.	SDM Appendix B: CM Guidelines, p. B-9
Constraint	Factors that will limit the project management team's options, or restrict the development process.	IEEE Std 1490-1998, IEEE Guide-Adoption of PMI Standard-A Guide to the Project Management Body of Knowledge, Section 4.1.1.1
Conversion Plan	Plan addressing all applicable installation strategies and conversion procedures for converting existing automated and manual systems, including system application software, support software, security software and hardware, telecommunication equipment, and peripheral devices at all sites (including pilot and production sites).	SDM Section 5: Evaluate System, p. 3, 13
Corrective Maintenance	Modifications that fix application problems caused by design, logic or coding errors.	SDM Appendix D: System Maintenance, p. D-3
Criticality	The degree of impact that a requirement, module, error, fault, failure, or other item has on the development or operation of a system.	IEEE Std 610.12-1990
Cycle	A period of time during which a set of events is completed. A set of operations that is repeated regularly in the same sequence, possibly with variations in each repetition.	IEEE Std 610.12-1990
Data Administration	Effective management of HUD's data resources. All systems are created and maintained in accordance with HUD's data administration policy and practices. Project activities are carried out consistent with the data management environment, and data administration concerns are addressed during all activities of the lifecycle.	SDM Introduction, p. 7

Word/Term	Definition	Reference
Data Management	In accordance to Federal guidelines, the effective handling of shared data across systems and programs, and the ensuring of data quality, clear definition of data, and proper safeguarding of systems data during design and implementation of automated and manual processes.	SDM Introduction, p. 9
Data Requirements Document	Documentation of the detailed description of the data model that the system must use to fulfill its functional requirements. The data is categorized, defined, the constraints defined, import responsibilities identified, and data collection requirements identified in this document.	SDM Section 3 Define System, p. 16
Define System Phase	Involves developing a detailed functional statement of the user's need and developing a project plan covering the estimated cost, schedule, and technical parameters of the project.	SDM Introduction, p. 5
Deliverable	A predetermined tangible and verifiable work product to be delivered to the acquirer, such as an Functional Requirements Document, Project Plan, user’s manuals, or final product. Deliverables may be self-contained or part of a larger system’s deliverables.	IEEE Std 1058-1998
Design Baseline	Culminates from activities performed during the Design System phase, whose main component is a system/subsystem specification that defines the overall system design in terms of its subsystems, the allocation of requirements to subsystems, and interfaces between subsystems and external systems.	SDM Appendix B: CM Guidelines, p. B-6
Design System Phase	Involves developing and approving detailed specifications to fulfill the stated functional requirements.	SDM Introduction, p. 5
Development Baseline	Generated during the Build System phase, defines the detailed structure of the system being implemented.	SDM Appendix B: CM Guidelines, p. B-6
Documentation Testing	Evaluation of documentation for content, clarity, and consistency. Content refers to the relevance and completeness of the documentation and its applicability to the system. Consistency refers to the maintenance of standards throughout the documentation, uniform terminology, and consistency with other documents. The instructions from the user documentation are checked for accuracy against actual operation on the system.	SDM Section 5: Evaluate System, p. 7
Evaluate System Phase	Involves testing of the system to ensure that the system functions as desired and certifying that the system has completed development.	SDM Introduction, p. 5
Evaluation of Request (EOR)	Document used to evaluate the project dates and cost requirements within the ARN or Needs Statement.	SDM Section 1: Initiate Project, p. 4

Word/Term	Definition	Reference
Evolutionary Development Lifecycle Model	Like the incremental development lifecycle, this model also develops a system in small releases, but differs with the incremental in that it acknowledges that users needs are not fully understood and not all requirements can be defined at the start, therefore user needs and system requirements are partially defined at the start, and then are refined in each succeeding small release. The system evolves as the understanding of user needs and the resolution of issues occurs. Prototyping is especially useful in this model.	SDM Appendix A: Project Management Guidelines, p. A-12
Feasibility Study	Description of the methodology and criteria used to determine the feasibility and the preferred approaches selected for fulfilling the system needs.	SDM Section 1: Initiate Project, p. 24
Functional Baseline	The main product of the Define System phase, whose main components are functional requirements and data requirements documents that describe the system's functional, data, performance, interface, reliability, and security requirements.	SDM Appendix B: CM Guidelines, p. B-6
Functional Requirement	A requirement that specifies a function that a system or system component must be able to perform.	IEEE Std 610.12-1990
Functional Requirements Document (FRD)	Documentation of initial definition of the system and the environment in which it will operate, along with requirements that must be associated with approved Technology Investment Board activities and within the scope of Initiate Project phase documents. It describes the current procedures for the system, if applicable, and the procedures for the proposed system; identifies organizational, operational, and user impacts; and determines assumptions and constraints.	SDM Section 2: Define System, p. 9
Functional Testing	The correct operation of an application; the correct addition, modification, and deletion of data from the system; and the correct operation of any audit trails in the system.	SDM Section 5: Evaluate System, p. 7
Incremental Development Lifecycle Model	A software development technique in which user needs are determined, system requirements defined, and the remaining steps of development are performed in small sequential releases, with the first incorporating part of the planned capabilities and more capabilities being added with each next small release until system completion.	SDM Appendix A: Project Management Guidelines, p. A-8
Independent Validation and Verification (IV&V)	Verification and validation performed by an organization that is technically, managerially, and financially independent of the development organization.	IEEE Std 610.12-1990
Initiate Project Phase	Involves identifying the information management need and deciding whether to commit the resources necessary to address the need.	SDM Introduction, p. 5
Interface Testing	Evaluation of the system’s ability to perform required interfaces with other systems operated by HUD and outside organizations.	SDM Section 5: Evaluate System, p. 7

Word/Term	Definition	Reference
Internal Audit Plan	Developed by the Inspector General's (IG's) staff, a review plan of systems under development, including sensitivity and criticality of the systems or the effectiveness of internal agency information system management control. The plan clarifies audit involvement, ranging from the review of the completed work product to active audit participation in each system development activity, the overall objective being to assess the adequacy of internal data processing control and providing reasonable assurance to management that adequate controls are in place.	SDM Section 2: Define System, p. 22.
Inventory Log	Log of software and hardware used in the development of a product with the following information: title of product, release date, version number or model number, name of the organization responsible for development of the product (usually the sponsoring organization), and the product distribution list.	SDM Section 5: Evaluate System, p. 21, 22
Lessons Learned	Historical database for projects that records the causes of variances, the reasoning behind the corrective action chosen, etc.	
Management Summary	One-page summary prepared for each product produced or revised during the Evaluate System phase, including a summary of essential data collected in a document product, conclusions that may be drawn from the document, and potential impacts on the project, if applicable.	SDM Section 5: Evaluate System, p. 4
Methodology	A comprehensive, integrated series of techniques or methods creating a general systems theory of how a class of thought-intensive work ought to be performed.	IEEE Std 730.1-1995
Milestone	A scheduled event used to measure progress, such as a deliverable.	IEEE Std 1058-1998
Needs Statement	Document(s) describing the deficiency and justifying the exploration of alternative solutions.	SDM Section 1: Initiate Project, p. 3
Operate System Phase	Involves installing the system at all sites and the onset of operational activities, including controls on all proposed computer hardware and software changes.	SDM Introduction, p. 5
Operational Testing	Testing of the procedures for installing and operating the systems software on its related hardware. It also tests the backup and recovery procedures.	SDM Section 5: Evaluate System, p. 7
Package-Based Development Lifecycle Model	System development based on the use of commercial off-the-shelf (COTS) and Government off-the-shelf (GOTS) products and reusable packages.	SDM Appendix A: Project Management Guidelines, p. A-14
Peer Review	A meeting of individuals (i.e., peers) for the purpose of performing a structured review of a work product so as to reduce the number of errors and improve the overall quality of the work product by finding errors early in the development of the work product and ensuring that appropriate standards are observed.	
Perfective Maintenance	Modifications that satisfy a need to improve the application in some area that is a nonfunctional requirement or to resolve a problem before the user community is affected.	SDM Appendix D: System Maintenance, p. D-4

Return to Title Page

Word/Term	Definition	Reference
Performance Testing	Testing showing whether the program satisfies its performance or efficiency requirements.	SDM Section 5: Evaluate System, p. 7
Pilot Test Report	Documentation of results from pilot system operation with records of any errors and feedback from the monitoring of the system.	SDM Section 6: Operate System, p. 8
Practice	Requirement employed to prescribe a disciplined uniform approach to a software development process.	IEEE Std 610.12-1990
Procedure	Ordered series of instructions that a user follows to do one or more tasks.	IEEE Std 1063-1987
Procedure Testing	Examination of the interface between the programs (system) and any manual systems or human procedures, such as those followed by the system operator, database administrator, or terminal user.	SDM Section 5: Evaluate System, p. 7
Process	A sequence of steps performed for a given purpose.	IEEE Std 610.12-1990
Product Baseline	Established during the Evaluate System phase. The product baseline's major component is the end system product as built by the developers. A second major component includes the system user's operations, and maintenance manuals and the system installation and conversion procedures.	SDM Appendix B: CM Guidelines, p. B-7
Program	A group of projects managed in a coordinated way to obtain benefits not available from managing them individually, and may contain ongoing operations.	IEEE Std 1490-1998, IEEE Guide-Adoption of PMI Standard-A Guide to the Project Management Body of Knowledge, Section 1.5
Program Review	Reviews that evaluate a program with respect to the project plan in the areas of cost, scheduling, and technical issues, and consist of: • A brief presentation to a panel of experts regarding the current project status based on planned milestones; • A detailed written report presenting information regarding the dollars spent, hours worked, and deliverables completed.	
Project Approvals	Formal approvals occurring at each phase in the lifecycle to confirm management support of the project and the resulting system. These approvals are not the goal of the lifecycle process, but key means to the desired end: successfully solving an information management need.	SDM Introduction, p. 7
Project Management	The planning and managing of a project which includes: defining the scope of the project; selecting the best lifecycle model; organizing the work in manageable units; estimating the time to complete each unit of work; assigning staff to perform each work unit; and organizing the project development team to include independent test, CM, Quality Assurance, and other support personnel.	SDM Introduction, p. 10
Project Management Review	Project status is evaluated relative to the applicable project plans, schedules, standards, and guidelines.	IEEE Std 12207.2-1997

Return to Title Page

Word/Term	Definition	Reference
Project Manager	Individual responsible for the project's success and works through a project team and other supporting organization structures (e.g., working groups and user groups) to accomplish the objectives of the project, and who is responsible for ensuring that project activities and decisions consider the perspectives of all affected.	SDM Introduction, p. 8
Project Plan	A plan is produced during the Initiate Project phase that is updated, expanded, and refined continually throughout system development; covers project scheduling, work breakdown structure (WBS), staffing, resources, adjustments to the system development cycle structure, selection of tools and techniques, identification of applicable reviews and approvals, CM methods, and other related topics; and subject to review and approval by HUD management.	SDM Introduction, p. 6, 8
Project Review	Review conducted at each phase of the system to ensure that the system ultimately established is programmatically and technically sound and that key issues are identified and addressed appropriately as early as possible in the system's development to avoid major, expensive rework in later activities. The review would also provide feedback to the project team.	SDM Introduction, p. 6
Project Schedule	Usually developed during or immediately after the acceptance of the initial Project Plan, this schedule includes at least the planned start and expected finish dates for each detail activity, such as milestones and other deliverables.	IEEE Std 1490-1998, IEEE Guide-Adoption of PMI Standard-A Guide to the Project Management Body of Knowledge, Section 6.4.3.1
Project Sponsor	Individual or organization that helps to ensure effective planning, management, and commitment to information system projects and serves in a leadership role, providing guidance to the project team and securing from senior management the required reviews and approvals at specific points in the lifecycle.	SDM Introduction, p. 8
Quality	The degree to which a system, component, or process meets specified requirements.	IEEE Std 610.12-1990
Quality Assurance (QA)	Establishing of plans, standards, and procedures that will add value to the software project, satisfy the constraints of the project and the HUD IT organization's policies and fit the project's needs, and are useful for performing reviews and audits throughout the software lifecycle; conducting continuous reviews project activities, audits software work products throughout the lifecycle, and provides management information with which to judge whether the software project is adhering to its established guidelines.	SDM Introduction, p. 7; SDM Section 1: Initiate Project, p. 4
Quality Assurance Measurement and Analysis	The collection of measurements to determine the cost and schedule of QA activities.	

Word/Term	Definition	Reference
Quality Assurance Plan	Plan defining responsibilities and authority of the QA group; resource requirements for the QA group (staffing, tools, facilities); schedule and funding of the project's QA group activities; evaluations of products an activities to be performed by the QA group (including operational and support software, deliverable and nondeliverable products, software and non-software products (e.g., documents), and product development and verification activities); audits and reviews; project standards and procedures to be used as the bases for the audits and reviews; procedures for documenting and tracking noncompliance issues to closure; and documentation the QA group is required to produce.	
Quality Assurance Program	Program ensures that the project is following approved HUD standards and procedures and quality is checked throughout the lifecycle.	
Quality Control	A set of activities designed to evaluate the quality of developed or manufactured products produced by others or oneself.	IEEE Std 610.12-1990
Requirement	A condition or capability needed by a user to solve a problem or achieve an objective. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.	IEEE Std 610.12-1990
Requirement Change (RC)	A request to change technical requirements approved for the project, usually generated by system users, affects the scope of the system, and may have an impact on the project's overall cost, schedule, and technical capability.	SDM Appendix B: CM Guidelines, p. B-9
Requirements Analysis	The process of studying and refining user needs to arrive at a definition of system, hardware, or software requirements.	IEEE Std 610.12-1990
Requirements Document	A document containing any combination of recommendations, requirements, or regulations to be met by a software package.	ISO/IEC 12119-1994
Requirements Validation	Class of testing that ensures that all functional requirements are implemented as originally envisioned by the users and that all requirements are accounted for in the system.	SDM Section 5: Evaluate System, p. 7

Word/Term	Definition	Reference
Risk Analysis	Identifying the strengths and weaknesses of the proposed project as well as alternatives for improvement, the purpose of which is to select cost-effective safeguards that will reduce security-related risks to an acceptable level. A risk analysis document would contain the following determinations: • Type of risk analysis; • Security requirements based on sensitivity and criticality; • Baseline security requirements that must be satisfied; • Threats and vulnerabilities that could be exploited; and • Countermeasures to the identified threats and vulnerabilities.	SDM Section 1: Initiate Project, p. 4
Small maintenance efforts	Activity consisting of quick reaction assignments that can be implemented and installed with one staff month or less of effort. The assignments can address an error (corrective), a cosmetic change (perfective), or a mandated enhancement (adaptive), and programmer concentrating on addressing the problem, not updating the documentation, unless needed by the user to use the software.	SDM Appendix D: System Maintenance, p. D-7
Software Development Life Cycle (SDLC)	The period of time that begins with the decision to develop a software product and ends when the software product is no longer available for use. The cycle typically consists of a concept phase, requirement phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and sometimes retirement phase.	IEEE Std 610.12-1990
Software Engineering Institute Capability Maturity Model (SEI CMM)	A framework that describes the key elements of an effective software development plan; discusses practices for planning, engineering, developing, and managing software development and maintenance projects; and describes an evolutionary improvement path from an ad hoc, immature process (Level 1) to a mature, disciplined process (Level 5).	SDM Introduction, p. 9
Spiral Lifecycle Model	A model of the software development process in which the constituent activities, typically requirements analysis, preliminary and detailed design, coding, integration, and testing, are performed iteratively until the software is complete.	IEEE Std 610.12-1990
Standards	Mandatory requirements employed to prescribe a disciplined uniform approach to software/system development, maintenance, and operation.	IEEE Std 730.1-1995
Stress Testing	Testing that determines the maximum capacity of the system, given user requirements for response time and throughput.	SDM Section 5: Evaluate System, p. 7
System	A set of interlinked units organized to accomplish one or several specific functions.	IEEE Std 1219-1998
System Baseline	A set of controlled documents describing the system at a specific point in the system lifecycle.	SDM Appendix B: CM Guidelines, p. B-5

Return to Title Page

Word/Term	Definition	Reference
System Decision Paper (SDP)	Record of all decisions made during the Initiate Project phase, including: • Projected schedule of activities for the system life cycle, by phase; • Actual or projected beginning and ending dates for each phase; and • Running total of the actual or projected development and operation costs, by phase, and totaled by fiscal year.	SDM Section 1: Initiate Project, p. 4
System Decision Paper (SDP)	Record of decisions made during development of the Initiate Project phase documents (i.e., Needs Statement, risk analysis, feasibility study, benefit/cost analysis) and review board's approval or dismissal to continue the development of a project.	SDM Section 1: Initiate Project, p. 33
System Development Methodology (SDM)	Outline of activities and tasks for development efforts based on established standards and guidelines determined by HUD.	
System Maintenance	The activities required to keep a system operational and responsive to user's changing needs after the system is accepted and placed into production.	SDM Appendix D: System Maintenance, p. D-3
Technical Review	Review to evaluate the software products or services under consideration and provide evidence that: • They are complete and comply with HUD standards and specifications; • Changes are properly implemented and affect only those areas identified by the CM Process; • Applicable schedules are adhered to; • They are ready for the next planned activity; • Development, operation, or maintenance is being conducted according to the plans, schedules, standards, and guidelines of the project.	IEEE Std 12207.2-1997
Technology Investment Board (TIB)	Responsible for ensuring that Departmental resources are directed to the highest Information Resource Management priorities of the Department and that these limited resources are efficiently utilized. The TIB tracks and reviews progress toward achieving key events and target dates for critical project activities.	SDM Section 1: Initiate Project, p. 4
Test Results and Evaluation Report	A documented compilation of test results and summary of the system's readiness for production.	SDM Section 5: Evaluate System, p. 3
Test Team	Individuals, independent of the software development organization, who are designated by the project sponsor to test systems in scenarios for acceptability.	SDM Section 5: Evaluate System, p. 3
User	Person who uses software/system to perform some task.	IEEE Std 1063-1987
User Manual	Documentation of user procedures, including the system description, input and expected output procedures, user interface descriptions, file query, and office procedures, so as to allow for the determination of the system's applicability and when and how to used it.	SDM Section 4: Build System, p. 19

Return to Title Page

Word/Term	Definition	Reference
Validation, Verification, &Test (VV&T) Plan	Documentation of the testing strategy that provides acceptance testing for all components of a system, including detailed requirements of all tests, testing methods and tools, and test evaluation criteria.	SDM Section 1: Design System, p. 18
Volume Testing	Testing intended to demonstrate the system's capability to handle the volume of data specified in the requirement.	SDM Section 5: Evaluate System, p. 7
Waterfall Development Lifecycle Model	A model of software development process which is essentially a once-only, sequential-step approach with the following sequential activities: determine user needs, define requirements, design the system, implement the system, test, fix and deliver the system.	SDM Appendix A: Project Management Guidelines, p. A-6
Work Breakdown Structure (WBS)	A structure of well defined work efforts and work elements that organizes and defines the total scope of the project and provides for better planning, scheduling, controlling of a project's work efforts.	SDM Appendix F: WBS, p. F-3